LIFTED INFERENCE FOR RELATIONAL HYBRID MODELS

BY

JAESIK CHOI

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Computer Science
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2012

Urbana, Illinois

Doctoral Committee:

Associate Professor Eyal Amir, Chair, Director of Research
Professor Dan Roth
Professor Steven M. Lavalle
Professor David Poole, University of British Columbia

ABSTRACT

Probabilistic Graphical Models (PGMs) promise to play a prominent role in many complex real-world systems. Probabilistic Relational Graphical Models (PRGMs) scale the representation and learning of PGMs. Answering questions using PRGMs enables many current and future applications, such as medical informatics, environmental engineering, financial forecasting and robot localizations. Scaling inference algorithms for large models is a key challenge for scaling up current applications and enabling future ones.

This thesis presents new insights into large-scale probabilistic graphical models. It provides fresh ideas for maintaining a compact structure when answering questions or inferences about large, continuous models. The insights result in a key contribution, the Lifted Relational Kalman filter (LRKF), an efficient estimation algorithm for large-scale linear dynamic systems. It shows that the new relational Kalman filter enables scaling the exact vanilla Kalman filter from 1,000 to 1,000,000,000 variables. Another key contribution of this thesis is that it proves that typically used probabilistic first-order languages, including Markov Logic Networks (MLNs) and First-Order Probabilistic Models (FOPMs), can be reduced to compact probabilistic graphical representations under reasonable conditions. Specifically, this thesis shows that aggregate operators and the existential quantification in the languages are accurately approximated by linear constraints in the Gaussian distribution. In general, probabilistic first-order languages are transformed into nonparametric variational models where lifted inference algorithms can efficiently solve inference problems.

To My Loving Family and Trusted Friends.

ACKNOWLEDGMENTS

Fortunately enough, pursuing my Ph.D. degree in Urbana-Champaign was the happiest time of my life. Inspiring fellow researchers guided my research. My loving family and trusted friends fully supported me during the studies.

First of all, I deeply appreciate my advisor, Eyal Amir, for his generous support during my studies. Without him, I may not have been able to work on such exciting research problems including my thesis topic. His invaulable advice and guidance were crucial to making progress on this thesis. He always has been a trustful mentor and has been with me almost like a longtime friend. He often treated himself as an equal collaborator instead of a supervisor, and encouraged me to lead the research projects. I appreciate his thoughtful and generous consideration more than I can say.

It was my pleasure to interact closely with Dan Roth during the studies. His gentle and wise advice helped me several times. His excellent machine learning courses provided valuable foundations of artificial intelligence. His emphasis on critical thinking has significantly influenced my research. Also, his constructive feedback on this thesis greatly helped me plan future work.

I enjoyed discussing my thesis with Steven LaValle. Whenever I visited his office to get advice on my ideas, he always provided lots of related work without hesitation. Especially, his suggestions helped me a lot when I worked on robot planning problems. Also, he shared his candid experience how he came up with his seminal work, and encouraged me to realize the potentials of my current research work.

I was grateful to discuss with David Poole on my thesis. After Eyal suggested

that I read Davids previous work, I was fascinated with his papers and results. Also, I was indebted to David because his excellent research work always gave me exciting new directions to pursue. It was my great pleasure that he recommended my work to be included in a IJCAI-11 tutorial and agreed to be on my thesis committee. I deeply appreciate his splendid suggestions on my thesis.

I also want to thank to Gerald Dejong for fruitful discussions on my thesis. His encouragement of my research toward human level intelligence gave me confidence on writing this thesis. I regret not discussing with him earlier.

In 2010, I spent an enjoyable summer with excellent researchers at SRI International. I appreciate Rodrigo de Salvo Braz and Hung Bui for active and energetic collaborations, as part of this thesis (Chapter 4) may not have existed without Rodrigo and Hung. Also, I thank Tuyen Ngoc Huynh and David Israel for valuable comments.

During my studies, I had exciting opportunities to work on real-world environmental problems. I thank David J. Hill, Yong Liu, Tiangfang Xu and Albert J. Valocchi for sharing important research problems and collaborating to solve them.

I thank Abner Guzman-Rivera. He has been an intimate friend. His different perspective stimulated me to advance my ideas. Also, his efforts substantially improved the writing of Chapter 3. Collaboration with him was very exciting experience to me. I will never forget the day in Barcelona when we revised the IJCAI-11 presentation slides all night. Thank you, Abner. I hope your studies go well, too.

In earlier stages of my studies (just after passing the qualifying exam), I met Won Jong Jeon and Sang-Chul Lee. We worked together on addressing video matching problems. The work was not included in this thesis because it is beyond the scope of this thesis. However, I learned the basics on how to conduct research from the two fellow researchers.

I also wish to thank the members of my research group, the General Intelligence Group (GIG) or previously the Knowledge Representation and Reasoning (KRR) Group: Hannaneh Hajishirzi, Afsaneh Shirazi, Mark Richards, Deepak

Ramanchandran, Tsvi Achler, Wen Pu, Juan Mancilla Caceres, Codruta Girlea, and Dafna Shahaf. They gave valuable comments on my research which helped improve my thesis. I also appreciate valuable comments given by friends in other artificial intelligence groups: Duan Tran, Ming-Wei Chang, Vivek Srikumar, Alexander Sorokin, Quang Do, Leonardo Bobadilla, and Varsha Hedau.

Throughout my studies, my department staff was very supportive and kind to me. I want to give special thanks to Donna Coleman, Mary Beth Kelley, and Keely Ashman.

My trusted friends continuously encouraged me to complete this thesis. Especially, I want to thank Ikkjin Ahn, Heeseok Kim, Jeongkeun Lee, Donghwan Jeon, Daehyun Yoon, and their families for listening to my stories and sharing their wisdom from the beginning of my Ph.D. studies.

Since I arrived at Urbana-Champaign, many Korean friends helped me settle down and gave me valuable feedback when I prepared important milestones including the qualifying exam, the preliminary exam and the thesis defense. I deeply appreciate them: Hyunsu Ju, Bongki Kim, Youngha Kim, Dongyun Jin, Eunsoo Seo, Sangkyum Kim, Steve Ko, Kihwal Lee, Jin Heo, Jungmin So, Wonson Ahn, Younhee Ko, Sungjin Im, MyungJoo Ham, Yoonkyong Lee, YoungMin Kwon, Han-Ul Yoon, Jihyuk Choi, Wucherl Yoo, Hyun Duk Kim, Keun Soo Yim, Minyoung Nam, Jung-Eun Kim, Man-Ki Yoon, Wooil Kim, and Minje Kim.

Many parts of my thesis are written at coffeehouses in Urbana-Champaign, Illinois and Menlo Park, California. I wish to thank baristas for making me tasteful cups of coffee at Starbucks or Expresso Royal cafes.

Before coming to Urabana-Champaign, I began my journey as a researcher at Korea Institute of Science and Technology with Woojin Chung. He is not only an excellent reseacher but also a great person. I really appreciate him for advising me on my first paper and giving practical guidance to enjoy Ph.D studies.

I cannot express enough gratitude to my family: my parents, Yongseon Choi and Kyunghye Seo, my parents in law, Donmo Sung and Insook Choi, my sister, Jiwon Choi, and my brother in law, Yeol-min Seong. My family always trusted

me and gave me unending support. I missed you all during the time of my studies. I hope to spend more time with you, now.

I am the luckiest man for being a husband of my wife, Jihye Seong. She is a lovely wife, a fabulous mother and an excellent researcher. She always has been with me and held my hands in good and bad. Whenever I had hard times, her positive mind embraced my heart and got me be back on track. Also, my son, Edward, and another soon-to-be-born son are sources of my energy. Thank you guys giving me endless happiness.

Throughout my studies, I was supported by grants given to me or my advisor. I wish to acknowledge the support of Korea Research Foundation and National Research Foundation of Korea award 2005-215-D00314 – Overseas Studies Grant, Beckman Institute and University of Illinois award CS/AI 2009 – Cognitive Science/Artificial Intelligence, National Science Foundation (NSF) award IIS-05-46663 – CAREER: Scaling Up First-Order Logical Reasoning with Graphical Structure, UIUC/NCSA AESIS initiative and 251024 grant, NSF award IIS-09-17123 – RI: Scaling Up Inference in Dynamic Systems with Logical Structure and NSF award ECS-09-43627 – Improving Prediction of Subsurface Flow and Transport through Exploratory Data Analysis and Complementary Modeling, and DARPA award FA8750-09-C-0181 - the DARPA Machine Reading Program under Air Force Research Laboratory (AFRL). I also want to give thanks for UAI-10, IJCAI-11 and AAAI-11 travel scholarships.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

AFM	Aggregate Factor Marginalization
DGM	Directed Gaussian Models
FOPM	First Order Probabilistic Model
FOVE	First Order Variable Elimination
GMRF	Gaussian Markov Random Field
HMM	Hidden Markov Model
KF	Kalman filter
LRKF	Lifted Relational Kalman filter
LRVI	Lifted Relational Variational Inference
MLN	Markov Logic Network
PGM	Probabilistic Graphical Model
RCM	Relational Continuous Model
RGM	Relational Gaussian Model
RHM	Relational Hybrid Model
RPM	Relational Pairwise Model
RM	Relational Model
ROM	Relational Observational Model
RTM	Relational Transition Models
PRGM	Probabilistic Relational Graphical Model
TVD	Total variation distance
WLOG	Without loss of generality

CHAPTER 1

INTRODUCTION

This thesis presents new insights and algorithms for Probabilistic Graphical Models (PGMs). A PGM is a graphical representation of a joint probability distribution over random variables. Each node in the graph represents a random variable. Each factor in the graph represents a joint probability over a subset of the random variables. By grouping random variables, Probabilistic Relational Graphical Models (PRGMs), or Relational Models (RMs), enable scaling up the representation and learning of PGMs. Inference, answering questions, with the relational models enables many current and future applications, such as medical informatics, environmental engineering, financial forecasting, and robot localizations. Scaling inference algorithms for large models is a key challenge to scaling up current applications and enabling future ones.

Inference, computing various probabilities of interest, with large PRGMs is hard because large graphs tend to include large cliques, sets of fully interconnected random variables. In general, the computational solution of an inference problem with such a model becomes exponentially harder as the number of random variables in the largest clique increases. Thus, many inference problems for large relational models are intractable.

This thesis delivers fresh insights into and algorithms for large-scale probabilistic graphical models, including clustered random variables. It presents new ideas that maintain a compact structure when solving inference problems for relational models with continuous random models. The insights expand to a key contribution, the Lifted Relational Kalman filter (LRKF), an efficient estimation algorithm for large-scale linear dynamic systems which shows that the LRKF enables scal-

ing the exact Kalman filter from 1,000 to 1,000,000,000 variables.

Another key contribution of this thesis is that it proves that regularly used probabilistic first-order languages, including Markov Logic Networks (MLNs) and First-Order Probabilistic Models (FOPMs), can be reduced to compact probabilistic graphical representations under reasonable conditions. Specifically, this thesis shows that aggregate factors and the existential quantification in the languages are accurately approximated by linear constraints in the Gaussian distribution. In general cases, variational models approximate PRGMs with bounded errors. Variational models also pave the way for solving inference problems efficiently. These advances have been directly applied in real-world groundwater models [Xu *et al.*, 2012]. The similar principles also have been applied to multiple domains in computer vision [Choi *et al.*, 2011c; 2008], robot planning [Choi and Amir, 2007; 2009], network abuse detection [Choi *et al.*, 2010b] and decision making [Hajishirzi *et al.*, 2009].

1.1 Inference with Relational Hybrid Models

1.1.1 Overview

Relational Hybrid Models (RHMs) represent relationships among sets of random variables with continuous and discrete domains in a concise manner. The intuition of RHMs is that each set of random variables has the same numbers and types of relationships as other sets. For example, prices of houses in the same residential district may change together. Two random variables, representing the prices of a house A and a neighboring house B, may have the same relationship with another random variable, the mortgage rate. Thus, one may also model the relationship with the same factor, e.g., having the same Gaussian noise.

In this thesis, probabilistic first-order language describes relationships among sets of discrete and continuous random variables for the RHMs. The probabilistic language handles uncertainty using probability theory and exploits structure

using first-order logic. The language provides an expressive formalism that represents the joint probability distribution of a large number of random variables. The language first defines a first-order logic sentence over the universe of random variables. Any set of random variables satisfied by the first-order logic sentence has the same factor over the set of random variables. In this way, the relational models can compactly represent the joint probability distribution without redundancies.

The language allows the utilization of the first-order structure for efficient inference. It is well known that first-order logic allows for efficient reasoning procedures by enumerating first-order logic sentences without referring to all propositional, or individual, elements. Lifted inference algorithms can calculate the conditional and marginal probabilities for RHMs by uplifting the model structures and referring only to first-order relationships, not all propositional variables.

1.1.2 What Is the Problem?

Many real-world systems in finance, environmental engineering, and robotics include continuous domains. One cannot avoid dealing with continuous random variables when answering questions about the systems. Unfortunately, most principles devised for discrete RMs, e.g. [Poole, 2003; de Salvo Braz *et al.*, 2005; Milch and Russell, 2006; Richardson and Domingos, 2006], are not applicable to such complex continuous systems and require discretizing continuous domains. Furthermore, discretization and usage of discrete lifted inference algorithms is highly imprecise. Therefore, the first fundamental challenge addressed in this thesis is building probabilistic representation languages and efficient inference algorithms for RMs with continuous variables.

Another key challenge is handling individual attributes of random variables in relational models. For example, an RM can represent a housing market model in a country. The models should be able to handle the price of each house in the country. However, most existing lifted inference algorithms force random variables to have the exact same attributes so they are in the same group. Thus, whenever new

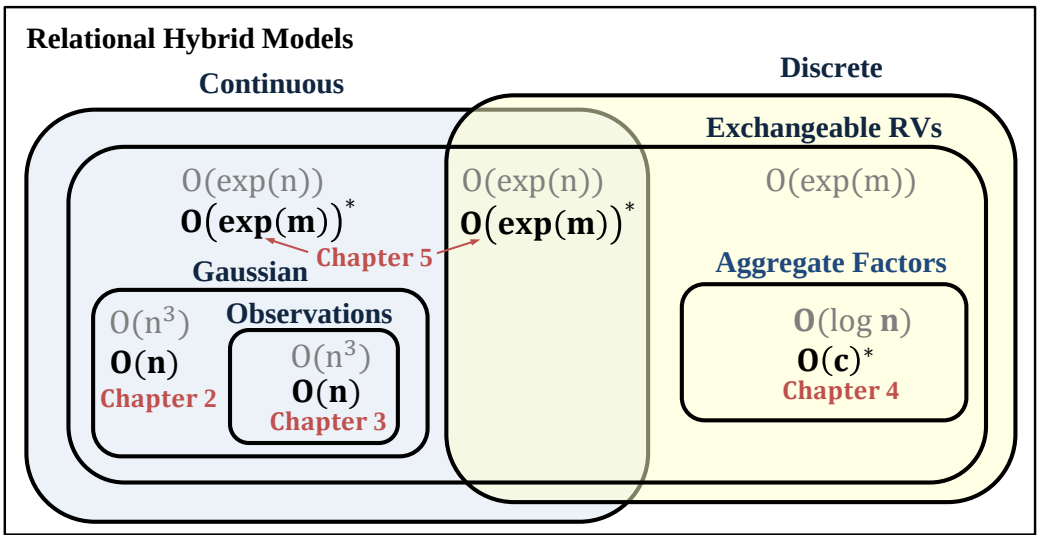


Figure 1.1: An illustration of the contributions of this thesis. The complexities displayed with bold fonts represent the new results of this thesis on solving inference problems with RHMs. Here, n is the number of random variables (RVs); m is the number of RV clusters; c is a constant; and $*$ represents an approximation. Aggregate factors and exchangeable RVs will be defined in the relevant chapters.

attributes are given or observed in an individual random variable, the attributes force the lifted inference algorithms to deteriorate the first-order structures into fine-grained propositional structures. Given such propositional structures, lifted inference algorithms refer to all ground random variables, and do as badly as propositional inference algorithms.

1.1.3 The Contributions of This Thesis

This thesis introduces new lifted inference algorithms that compute the conditional (or marginal) probability of RHMs. The first contribution (Chapter 2) is a new lifted inference algorithm for RMs with only continuous variables, or Relational Continuous Models (RCMs). The algorithm maintains relational structures during the inference procedure for relational pair-wise potentials, such as pairwise linear Gaussian potentials.

The second contribution (Chapter 3) is an efficient exact filtering algorithm,

the LRKF, for large-scale linear dynamic systems. In each time step, the lifted inference algorithm efficiently updates a large number of random variables. The LRKF maintains compact pairwise relationships among random variables even with individual observations. Thus, individual attributes do not degenerate the relational structures into propositional ones.

The third contribution (Chapter 4) is a new insight into aggregate operations in RMs[1]. It shows that aggregate operators over relational models can be accurately approximated using linear constraints over Gaussian distributions. Thus, in many cases, calculating the conditional probability does not depend on the number of random variables. The accuracy of approximation is close to optimal when the model has a large number of random variables.

The last contribution (Chapter 5) includes new variational models, which present a new understanding of RHMs and variational models. One of the key understandings is that some potentials over sets of random variables in RHMs can be represented by a mixture of a joint probability distribution over independent and identically distributed (i.i.d.) random variables. The variational models seamlessly represent the discrete and continuous variables in the unified framework.

1.2 The Technical Results of This Thesis

1.2.1 Efficient Inference with Relational Continuous Models

Calculating a marginal over variables of interest is a typical inference task. At a propositional level, inference with a large number of continuous variables is non-trivial. Suppose that a random variable representing the market index such as S&P 500 depends directly on n random variables, revenues of n banks. When marginalizing the market index variable out, the marginal is a function of n variables (revenues of banks), thus marginalizing out remaining variables becomes

[1] Aggregate operators in RMs are equivalent to the existential quantification the probabilistic first-order languages

harder. When n grows, the computation becomes expensive. For example, when relations among variables follow Gaussian distributions, the computational complexity of the inference problem is $O(|U|^3)$ (U is a set of random variables). It limits the uses of such relational models to many large-scale real-world applications.

To address these issues, Probabilistic Relational Models (PRMs) [Ng and Subrahmanian, 1992; Koller and Pfeffer, 1997; Pfeffer *et al.*, 1999; Friedman *et al.*, 1999; Poole, 2003; de Salvo Braz *et al.*, 2005; Richardson and Domingos, 2006; Milch and Russell, 2006; Getoor and Taskar, 2007] describe probability distributions at a relational level with the purpose of capturing larger models. PRMs combine probability theory for handling uncertainty and relational models for representing system structures compactly. Thus, they facilitate construction and learning of probabilistic models for large systems. Recently, [Poole, 2003; de Salvo Braz *et al.*, 2005; Milch *et al.*, 2008; Singla and Domingos, 2008] showed that such models enable more efficient inference than possible with propositional graphical models, when inference occurs directly at the relational level.

Present exact lifted inference algorithms [Poole, 2003; de Salvo Braz *et al.*, 2006; Milch *et al.*, 2008] and those developed in the efforts above are suitable for discrete domains, thus can in theory be applied to continuous domains through discretization. However, the precision of discretizations deteriorates exponentially in the number of dimensions in the model, and the number of dimensions in relational models is the number of ground random variables. Thus, discretization and usage of discrete lifted inference algorithms is highly imprecise.

Here, this thesis presents the first exact lifted inference algorithm for Relational Continuous Models (RCMs), a new probabilistic first-order language for continuous domains. The main insight is that, for some classes of potential functions (or potentials), marginalizing out a ground random variable in a RCM can yield a RCM representation that does not force other random variables to become propositional. Further, relational pairwise models, i.e. products of relational potentials of arity 2, remain relational pairwise models after eliminating out ground random

variables in those models. Thus, it leads to the compact representations and the efficient computations. I report Gaussian potentials, which satisfy the conditions for relational pairwise models.

This thesis also adapts principles of Inversion Elimination, a method devised by [Poole, 2003], to continuous models. Inversion Elimination's step essentially takes advantage of an ability to exchange sums and products. The lifted exchange of sums and products translates directly to continuous domains. This is a unique approach to continuous models.

Given a RCM, the suggested algorithm marginalizes continuous variables by analytically integrating out random variables except query variables. It does so by finding a variable, and eliminating it by Inversion Elimination. If such elimination is not possible, Relational Atom Elimination eliminates each pairwise form in a linear time. If the marginal is not in pairwise form, it converts the marginal into a pairwise form.

1.2.2 The Lifted Relational Kalman Filtering

The Kalman filter (KF) [Kalman, 1960] accurately estimates the state of a dynamic system given a sequence of control-inputs and observations. It has been applied in a broad range of domains which include weather forecasting [Burgers *et al.*, 1998], localization and tracking in robotics [Limketkai *et al.*, 2005], economic forecasting [Bahmani-Oskooee and Brown, 2004] and many others. Given a sequence of observations and Gaussian dependences between variables, the filtering problem is to calculate the conditional probability density of the state variables at each timestep. Unfortunately, the KF computations are cubic in the number of random variables, which limits the use of the KF exact methods to domains with a limited number of random variables. This has led to the combination of approximation and sampling (e.g. the Ensemble Kalman filter [Evensen, 1994]).

The LRKF leverages the power of relational languages [Friedman *et al.*, 1999; Poole, 2003; Richardson and Domingos, 2006] to describe models of which rep-

resentations are independent of the size of populations involved. Various lifted inference algorithms for relational models have been proposed [Poole, 2003; de Salvo Braz *et al.*, 2005; Milch and Russell, 2006; Richardson and Domingos, 2006; Wang and Domingos, 2008; Choi *et al.*, 2010a]. These seek to achieve carry computations in time independent of the size of the populations involved.

However, the key challenge in relational filtering (of dynamic systems) is ensuring that the representation does not deteriorate to the ground case when multiple observations are made. As more observations are received, an increasing number of objects become distinguished. This precludes the use of previously known algorithms unless approximately equivalent objects are grouped with expensive clustering algorithms.

This thesis presents Relational Gaussian Models (RGMs) to model dynamic systems of a large number of variables in a relational fashion. RGMs have as their main building block the pairwise linear Gaussian potential as detailed in Section 3.2. Further, it proposes a new lifted filtering algorithm that marginalizes out random variables of the previous timestep efficiently, in time linear in the number of random variables, while maintaining the relational (RGM) representation.

This prevents the relational pairwise structure from being increasingly grounded even when individual observations are made for all random variables. Moreover, updating the relational model takes only quadratic in the number of relational atoms (sets of random variables).

One key insight is that, given identical observation models, even when the means of the random variables are dispersed their variances remain identical. This is sufficient to sustain a relational representation.

1.2.3 Efficient Inference with Aggregate Factors in Relational Models

Relational models can compactly (that is, intensionally) represent graphical models involving a large number of random variables, each of them representing a rela-

tion between objects in a domain [Koller and Pfeffer, 1997; Friedman *et al.*, 1999; Milch *et al.*, 2005; Richardson and Domingos, 2006].

While it is possible to take advantage of compactness only for representation and expand the model into a propositional (extensional) form for inference, lifted inference methods try to keep the representation as compact as possible even during inference, increasing efficiency [Poole, 2003; de Salvo Braz *et al.*, 2007; Milch *et al.*, 2008; Singla and Domingos, 2008] .

The first proposed lifted inference solutions could deal only with factors on a fixed number of random variables. *Aggregate* parametric factors (based on aggregate functions such as *OR*, *MAX*, *AND*, *SUM*, *AVERAGE*, *MODE* and *MEDIAN*), which are defined on a varying, intensionally defined set of random variables, still needed to be treated propositionally, with cost exponential in the number n of random variables.

[Kisynski and Poole, 2009] introduced lifted methods for aggregate factors that reduce this complexity to $O(rk \log n)$ for commutative associative aggregate functions on n k-valued random variables being aggregated into an r-valued random variable (and even $O(rk)$ for *OR* and *MAX*)[2]. However, for general cases (such as the non-associative function *MODE*), their exact inference process has time $O(rn^k)$, that is, polynomial in n.

Here, the contributions of this chapter are threefold. It contributes an *exact* solution *constant* in n when $k = 2$ for aggregate operations *AND*, *OR*, *MAX* and *SUM*.

It also presents an efficient (*constant* in n) approximate algorithm for inference with aggregate factors, for all typical aggregate functions.

The potential of an aggregate factor for a valuation v of a set of random variables depends only on the *histogram* on the distribution of k values in V (in what [Milch *et al.*, 2008] calls a *counting formula*).

This chapter shows that the typical aggregate functions but for *XOR*[3] can be

[2]Note that r=n for aggregate functions such as *SUM* of n binary variables.

[3]*XOR* has its own straightforward solution.

represented by linear constraints in the space of histograms (a $(k-1)$-simplex).

Because aggregate factors' potentials on the space of histograms can be approximated by a normal distribution, one can approximately sums over them (which is the main inference operation) by computing the volume under normal distributions truncated by linear constraints. This holds even for *MODE*, which is commutative but not associative.

This approximation can be computed analytically for all operations on binary random variables and for certain operations on multivalued ($k>2$) random variables such as *SUM* and *MEDIAN*. Otherwise, it is computed by Gibbs sampling with a limited number of iterations [Geweke, 1991; Damien and Walker, 2001]. Finally, a third contribution is a further optimization for aggregations of multiple groups of random variables, each with its own distribution.

1.2.4 Lifted Relational Variational Models

Many real-world systems can be described using continuous and discrete variables with relations among them. Such examples include measurements in environmental sensor networks, localizations in robotics, and economic forecasting in finance. In such large systems, efficient and precise inference is necessary. As an example from environmental science, an inference algorithm can predict a posterior of unobserved groundwater levels and contamination levels at different locations, and making such an inference precisely is critical to decision makers.

Probabilistic Relational Models (PRMs) [Ng and Subrahmanian, 1992; Pfeffer *et al.*, 1999; Friedman *et al.*, 1999; Richardson and Domingos, 2006] describe probability distributions at a relational level with the purpose of capturing the structure of larger models. These compact representations can facilitate the construction and learning of probabilistic models for large systems. A key challenge of inference procedures with RPLs is that they often result in intermediate density functions involving many random variables and complex relationship among them.

Real-world systems have large numbers of variables including both discrete and continuous. PRMs represent such large systems compactly. Lifted inference presently can address discrete models and continuous models, but not hybrid ones. For (d-valued) discrete variables, lifted inference can take advantage of the insight which groups equivalent models into a histogram representation with an order of $poly(d)$ entries [de Salvo Braz *et al.*, 2005; Milch and Russell, 2006; Jha *et al.*, 2010] (instead of $exp(d)$ entries in traditional *ground* models). For Gaussian potentials, lifted inference algorithms can maintain a compact covariance matrix during (and after) inference, e.g. [Choi *et al.*, 2010a; 2011b].

Unfortunately, these principles are not applicable to general (non-Gaussian) hybrid models because the histogram is not applicable to continuous domains without discretizations, and the covariance matrix is a special structure for Gaussians. Thus, existing variational methods, e.g. Latent Tree Models [Zhang, 2002; Choi *et al.*, 2011d] and Nonparametric Bayesian Logic [Carbonetto *et al.*, 2005], focus either on discrete models or Gaussian models.

This thesis provides a new insight (relational variational-inference lemmas) which accurately factors densities of relational models into mixtures of i.i.d. random variables. These lemmas enable us to build a variational approximation algorithm that takes large-scale graphical models with hybrid variables and finds close-to-optimal relational variational models. Then, lifted inference algorithms, a variable elimination and a Markov chain Monte Carlo (MCMC) sampling method, efficiently solve marginal inference problems on the variational models. This thesis shows that the algorithm gives a better solution than previous ones.

1.3 Plan of This Thesis

The main contributions of the thesis are included in Chapters 2, 3, 4 and 5. Chapter 2 formally defines RCMs, which are relational models with pairwise continuous potentials. Then, it presents an efficient inference algorithm for pairwise Gaussian

potentials. Chapter 3 extends the algorithm of Chapter 2 to create the LRKF, a new relational Kalman filter for relational linear dynamic systems. Chapter 4 presents lifted inference algorithms with aggregate factors. Chapter 5 presents a unified framework for relational hybrid models with the perspective of variational models. Chapter 6 summarizes the thesis and suggests opportunities for future studies.

Each chapter is written in an independent manner without assuming that readers have any background knowledge in relational models. Thus, it should be comprehensible to readers in computer science and other fields of science and engineering. Readers who are interested in Kalman Filtering should refer to Chapter 3. Those who are interested in variational inference should refer to Chapter 5.

1.4 Publication Notes

Below is the list of publications and chapters where they are revised and used :

- [Choi *et al.*, 2010a]: Chapter 2
- [Choi *et al.*, 2011b]: Chapter 3
- [Choi *et al.*, 2011a]: Chapter 4
- [Choi and Amir, 2011; 2012]: Chapter 5

CHAPTER 2

LIFTED INFERENCE FOR RELATIONAL CONTINUOUS MODELS

Relational Continuous Models (RCMs) represent joint probability densities over attributes of objects, when the attributes have continuous domains. With relational representations, they can model joint probability distributions over large numbers of variables compactly in a natural way. This section presents a new exact lifted inference algorithm for RCMs, thus it scales up to large models of real world applications. The algorithm applies to *Relational Pairwise Models* which are (relational) products of potentials of arity 2. Our algorithm is unique in two ways. First, it substantially improves the efficiency of lifted inference with variables of continuous domains. When a relational model has Gaussian potentials, it takes only linear-time compared to cubic time of previous methods. Second, it is the first exact inference algorithm which handles RCMs in a lifted way. The algorithm is illustrated over an example from econometrics. Experimental results show that our algorithm outperforms both a ground-level inference algorithm and an algorithm built with previously-known lifted methods.

2.1 Introduction

Many real world systems are described by continuous variables and relations among them. Such systems include measurements in environmental-sensors networks [Hill *et al.*, 2009], localizations in robotics [Limketkai *et al.*, 2005], and economic forecastings in finance [Niemira and Saaty, 2004]. Once a relational model among variables is given, inference algorithms can solve value prediction problems and classification problems.

At a ground level, inference with a large number of continuous variables is non-trivial. Typically, inference is the task of calculating a marginal over variables of interest. Suppose that a market index has a relationship with n variables, revenues of n banks. When marginalizing out the market index, the marginal is a function of n variables (revenues of banks), thus marginalizing out remaining variables becomes harder. When n grows, the computation becomes expensive. For example, when relations among variables follow Gaussian distributions, the computational complexity of the inference problem is $O(|U|^3)$ (U is a set of ground variables). Thus, the computation with such models is limited to moderate-size models, preventing its use in the many large, real-world applications.

To address these issues, Probabilistic Relational Models (PRMs) [Ng and Subrahmanian, 1992; Koller and Pfeffer, 1997; Pfeffer *et al.*, 1999; Friedman *et al.*, 1999; Poole, 2003; de Salvo Braz *et al.*, 2005; Milch *et al.*, 2005; Richardson and Domingos, 2006; Milch and Russell, 2006; Getoor and Taskar, 2007] describe probability distributions at a relational level with the purpose of capturing larger models. PRMs combine probability theory for handling uncertainty and relational models for representing system structures. Thus, they facilitate construction and learning of probabilistic models for large systems. Recently, [Poole, 2003; de Salvo Braz *et al.*, 2005; Milch *et al.*, 2008; Singla and Domingos, 2008] showed that such models enable more efficient inference than possible with propositional graphical models, when inference occurs directly at the relational level.

Present exact lifted inference algorithms [Poole, 2003; de Salvo Braz *et al.*, 2006; Milch *et al.*, 2008] and those developed in the efforts above are suitable for discrete domains, thus can in theory be applied to continuous domains through discretization. However, the precision of discretizations deteriorates exponentially in the number of dimensions in the model, and the number of dimensions in relational models is the number of ground random variables. Thus, discretization and usage of discrete lifted inference algorithms is highly imprecise.

Here, we propose the first exact lifted inference algorithm for Relational Continuous Models (RCMs), a new relational probabilistic language for continuous

domains. Our main insight is that, for some classes of potential functions (or potentials), marginalizing out a ground random variable in a RCM can yield a RCM representation that does not force other random variables to become propositional (Section 2.4). Further, relational pairwise models, i.e. products of relational potentials of arity 2, remain relational pairwise models after eliminating out ground random variables in those models. Thus, it leads to the compact representations and the efficient computations. We report Gaussian potentials which satisfy the conditions for relational pairwise models (Section 2.5). However, we are unsure whether the conditions are only satisfied by Gaussian potentials, yet.

We also adapt principles of *Inversion Elimination*, a method devised by [Poole, 2003], to continuous models. *Inversion Elimination*'s step essentially takes advantage of an ability to exchange sums and products. The lifted exchange of sums and products translates directly to continuous domains. This is a unique approach to continuous models, even though the insight is brought from discrete models.

Given a RCM, our algorithm marginalizes continuous variables by analytically integrating out random variables except query variables. It does so by finding a variable, and eliminating it by *Inversion Elimination*. If such elimination is not possible, *Relational Atom Elimination* eliminates each pairwise form in a linear time. If the marginal is not in pairwise form, it converts the marginal into a pairwise form.

This chapter is organized as follows. Section 2.2 provides the formal definition of RCMs. Section 2.3 overviews our inference algorithms. Section 2.4 presents main intuitions and results in a Gaussian potential. Section 2.5 provides the generalized algorithm for relational pairwise models. Section 2.7 provides experimental results followed by related works in Section 2.6. It concludes in Section 2.8.

2.2 Relational Continuous Models (RCMs)

We present a new relational model for continuous variables, Relational Continuous Models (RCMs). Relations among attributes of objects are represented by *Parfactor* models. [1] Each *parfactor* (L, C, A_R, ϕ) is composed of a set of logical variables (L)[2], constraints on L (C), a list attributes of objects (A_R), and a potential on A_R (ϕ). Here, each attribute is a random variable with a continuous domain.

We define a *Relational Atom* to refer the set of ground attributes compactly. For example, *Revenue*$[B]$ is a *relational atom* which refers to revenues of banks (e.g. $B = \{$'Pacific Bank', 'Central Bank', $\cdots\}$). To make the *parfactor* compact, a list of relational atoms is used for A_R. To refer to an individual random variable, we use a substitution θ. For example, if a substitution ($B =$ 'Pacific Bank') is applied to a relational atom, then the relational atom *Revenue*$[B]$ becomes a ground variable *Revenue*('Pacific Bank').[3] Formally, applying a substitution θ to a parfactor $g = (L, C, A_R, \phi)$ yields a new parfactor $g\theta = (L', C\theta, A_R\theta, \phi)$, where L' is obtained by renaming the variables in L according to θ. If θ is a ground substitution, $g\theta$ is a factor. Θ_g is a set of all substitution for a parfactor g. The set of *groundings* of a parfactor g is represented as $gr(g) = \{g\theta : \theta \in \Theta_{gr(L:C)}\}$. We use $RV(X)$ to enumerate the random variables in the relational atom X. Formally, $RV(\alpha) = \{\alpha[\theta] : \theta \in gr(L)\}$. $LV(g)$ refers the set of logical variables (L) in g.

The joint probability density over random variables is defined by *factors* in a *parfactor*. A *factor* f is composed of A_g and ϕ. A_g is a list of ground random variables (i.e. $(X_1(\theta), \cdots, X_N(\theta))$). ϕ is a *potential* on A_g: a function from $range(A_g) = \{range(X_1(\theta)) \times \cdots \times range(X_N(\theta))\}$ to non-negative real numbers. The factor f defines a weighting function on a valuation ($v = (v_1, \cdots, v_m)$): $w_f(v) = \phi(v_1, \cdots, v_m)$). The weighting function for a *parfactor* F is the product of weighting function of all factors, $w_F(v) = \prod_{f \in F} w_f(v)$. When G is a set of

[1] Part of its representation and terms are based on the previous works [Poole, 2003; de Salvo Braz *et al.*, 2005; Milch and Russell, 2006]. However, our representaion allows continuous random variables.

[2] Instead of objects, we use the general term, logical variables.

[3] *Revenue*() refers a random variable. *Revenue*[] refers a relational atom.

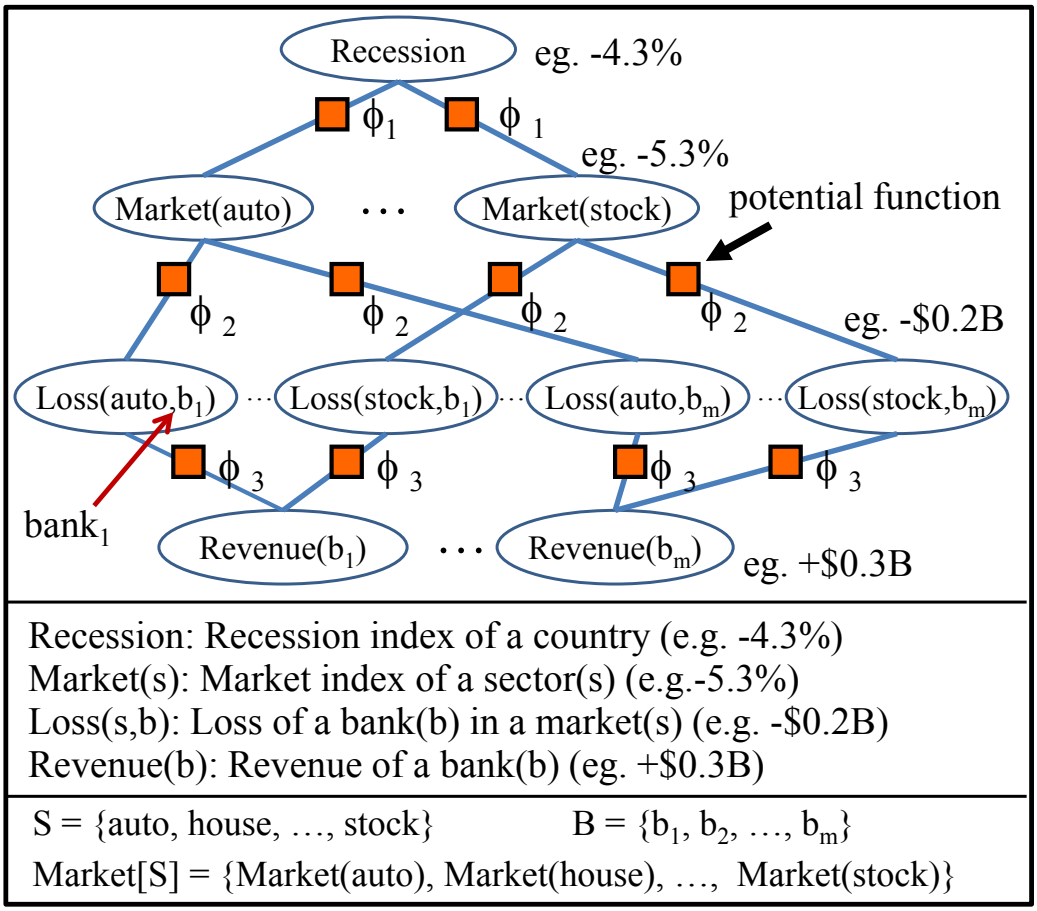


Figure 2.1: This figure shows a model among banks and market indices. *Recession* is a random variable. *Market[S]*, *Gain[S,B]* and *Revenue[B]* are relational atoms. The variable and atoms have continuous domain $[-\infty, \infty]$. For example, *Market(stock)* is -5.3%, and *Loss(stock,*B_m*)* is $-\$0.2B$.

parfactors, the density is the product of all factors in G:

$$w_G(v) = \prod_{g \in G} \prod_{f \in gr(G)} w_f(v). \tag{2.1}$$

For example, consider the model in Figure 2.1. S and B in L are two logical variables which represent markets and banks respectively. For example, S can be substituted by a specific market sector (e.g. S = *'stock'*). A parfactor $f_1 = (\{Market[S], Gain[S,B]\}, \phi_2)$ is defined over two relational atoms, $Market[S]$ and $Gain[S,B]$. $Market(s)$ (one variable in $Market[S]$) represents the quarterly market change (e.g. *Market(auto)*$=-3.1\%$). $Gain(s,b)$ represents the gain of bank

b in the market s. Given two values, a potential $\phi_1(Market(s), Gain(s,b))$ provides a numerical value. Given all valuations of random variables, the product of potentials is the probability density.

2.3 Algorithm Overview for RCMs

RCMs model large real-world systems in a compact way. One inference task with such models is to find the conditional density of query variables given observations of some variables.

PROCEDURE **FOVE-Continuous**(G,Q)
G: parfactors, Q: random variables (the query).

1. **If** $RV(G) = Q$ return G
2. $G \leftarrow$ **SPLIT**(G,Q)
3. $E \leftarrow$ **FIND-ELIMINABLE**(G,Q)
4. $G_E \leftarrow \{g \in G : RV(g)$ and $RV(E)$ intersect $\}$
5. $G_{\overline{E}} \leftarrow G \setminus G_E$
6. $g' \leftarrow$ **ELIMINATE-CONTINUOUS**(G_E,E) (Sections 2.4 and 2.5)
7. $G' \leftarrow \{g'\} \bigcup G_{\overline{E}}$
8. return **FOVE-Continuous**(G',Q)

PROCEDURE **ELIMINATE-CONTINUOUS**(G,E)
G: parfactors, E: a random variable to be eliminated

1. $g \leftarrow (LV(A_G \setminus E), C_G, A_G \setminus E, \prod_{g \in G} \Phi_g^{\frac{|\Theta_G|}{|\Theta_g|}})$
2. **If** (LV(E)=LV(g))
   return **Inversion-Elimination(g,E)**
   Else return **Relational-Atom-Elimination(g,E)**

PROCEDURE **FIND-ELIMINABLE**(G,Q)
G: parfactors, Q: $\subset RV(G)$ (G is split against Q)

1. For e from $A_G \setminus Q$
   $G_e \leftarrow \{g \in G : RV(g)$ and $RV(e)$ intersect $\}$
   If $LV(e) = LV(G_e)$ return e (for Inversion-Eliminable)
2. Choose e from $A_G \setminus Q$
3. return e (for Relational-Atom-Elimination)

Figure 2.2: FOVE_Continuous (First-Order Variable Elimination with continuous variables) algorithm.

Our inference algorithm, FOVE-Continuous (First-Order Variable Elimination), for RCMs recursively eliminates relational atoms. First, it *splits* (terminology

of [Poole, 2003]; *shattering* in [de Salvo Braz *et al.*, 2005])[4] relational atoms. The *split* operation makes groundings (e.g. $RV(X)$ $RV(Y)$) of every relational atoms (e.g. X Y) disjoint. It introduces observations as observations of groundings of separate relational variables. For example, observing $Market(auto) = 30\%$ creates two separate relational atoms: $Market(auto)$, $Market(M)_{M \neq auto}$. The '$M \neq auto$' then appears in parfactors relating to the latter relational atom. After *split*, *FIND-ELIMINABLE* finds a relational atom which satisfies conditions for one of the elimination algorithms: *Inversion-Elimination* (Section 2.5.2) and *Relational-Atom-Elimination* (Section 2.5.3). The found atom is eliminated by our *ELIMINATE-CONTINUOUS* algorithm explained in Sections 2.4 and 2.5. It iterates the elimination until only query variables are remained. The procedure is described in Figure 2.2.

Our main contributions are focused on the algorithm **ELIMINATE-CONTINUOUS**, a lifted variable eliminations for continuous variables. We describe details in Sections 2.4 and 2.5.

2.4 Inference with Gaussian Potentials

This section presents our first main technical contribution, efficient variable elimination algorithms for relational Gaussian models. We focus on the inference problem of computing the posterior of query variables given observations. It is important to efficiently integrating out relational atoms (e.g. Revenue[B] = $\{Revenue(b_1), \cdots, Revenue(b_m)\}$) for solving this inference problem.

In the following description, we omit the (inequality between logical variables and objects) constraints from parfactors. This allows us to focus on the potential functions inside those parfactors. The treatment below holds with little change for parfactors with such constraints.

[4]Please refer [Poole, 2003; de Salvo Braz *et al.*, 2005] for further details.

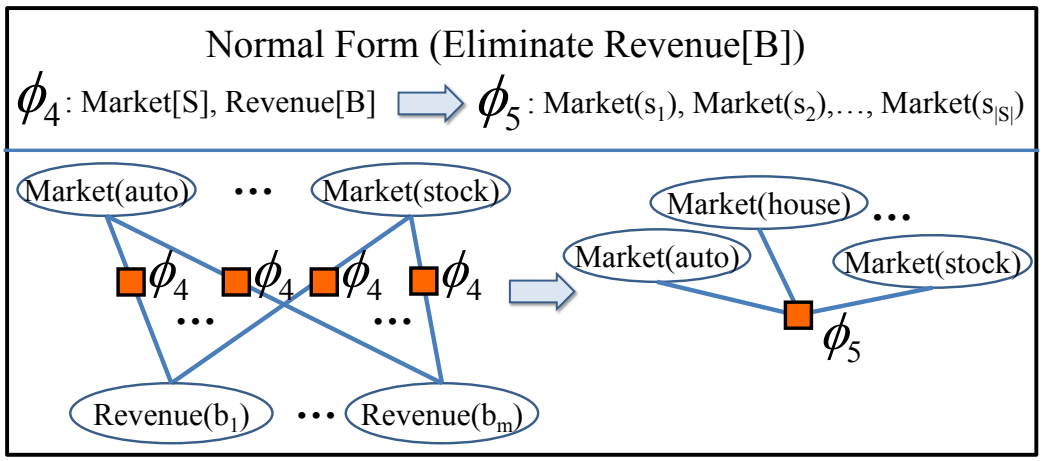


Figure 2.3: This figure shows a challenging problem in a RCM when eliminating a set of variables (*Revenue[B]*). Eliminating *Revenue[B]* in ϕ_4 generates an integral ϕ_5 that makes all variables in *Market[S]* ground. Thus, the elimination makes the RCM into a ground network.

2.4.1 Relational Pairwise Potentials

This section focuses on the product of potentials which we call *Relational Normals* (RNs). A RN is the following function with arity 2 (Section 2.5 provides a generalization for arbitrary potentials).:

$$\phi_{RN}(X,Y) = \prod_{x \in X, y \in Y} \frac{1}{\sigma\sqrt{2\pi}} exp\left(-\frac{(x-y)^2}{2\sigma^2}\right)$$

This potential indicates that the difference between two random variables follows Gaussian distributions.

Consider the models shown in Figure 2.3 and 2.4. The models represent the relationships between each market change and revenue of each bank. To simplify notations, we respectively shorten $Market(s)$, $Gain(s,b)$ and $Revenue(b)$ to $M(s)$, $G(s,b)$ and $R(b)$. The potential ϕ_4 in these figures is $\phi_{RN}(M(s),R(b))$, and the product of potential is $\prod_{s \in S, b \in B} \phi_{RN}(M(s),R(b))$

Figure 2.4 shows that integrating out a random variable $R(b_i)$ from the joint

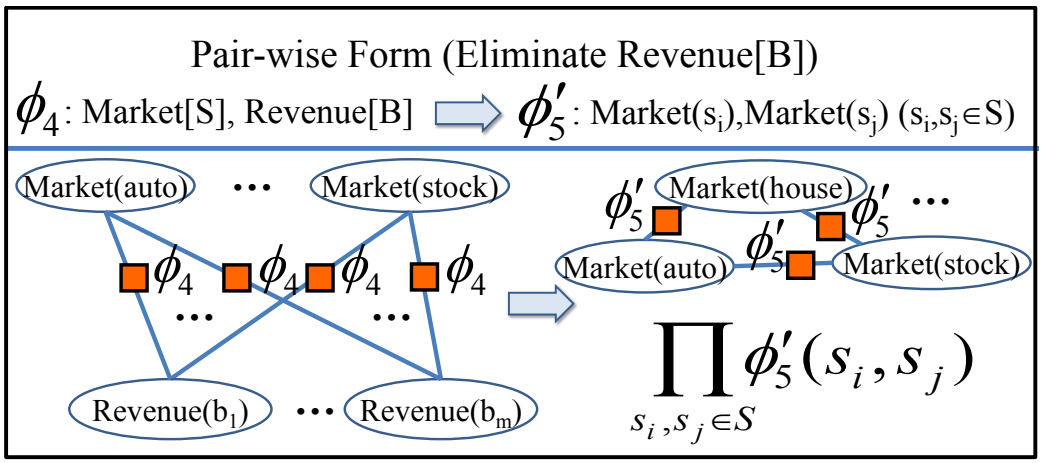


Figure 2.4: This figure shows our method for the problem shown in Figure 2.3. When eliminating *Revenue[B]*, we do not generate a ground network. Instead, we directly generate the pairwise form which allows the inference at the lifted level.

density results in the product of RNs again (c and c' are constants) as follow.

$$\int_{R(b_i)} \prod_{s \in S} \phi_4(M(s), R(b_i)) = c \cdot \exp\left(\frac{(\sum_{s \in S} M(s))^2}{2\sigma^2 \cdot |S|} - \frac{\sum_{s \in S} M(s)^2}{2\sigma^2}\right)$$
$$= c \cdot \prod_{1 \le i < j \le |S|} exp\left(-\frac{(M(s_i) - M(s_j))^2}{2\sigma^2 \cdot |S|}\right) = c' \cdot \prod_{1 \le i < j \le |S|} \phi'_5(M(s_i), M(s_j)) \quad (2.2)$$

Note that, following equations holds for integration.

$$\int_{\mathbf{R(b_i)}} \exp\left(-a\mathbf{R(b_i)}^2 + 2b\mathbf{R(b_i)} + c\right) = \sqrt{\frac{\pi}{a}} exp\left(\frac{b^2}{a} + c\right) \quad (2.3)$$

Here, the terms a and b can include random variables except $R(b_i)$.

Definition 1 (*Connected Relational Normal*) *The product of RNs is connected, when the connectivity graph is a connected component. Each vertex of the connectivity graph is a random variable or a constant in RNs, and each edge is a potential (RN).* ■

Lemma 1 *The product of RNs is a probability density function when it is connected, and at least a RN includes a constant argument.*

The proof is provided in Section 2.9.

2.4.2 Constant Time Relational Atom Eliminations

We provide two constant time elimination algorithms for RNs involving a single relational potential ϕ (i.e. the product of potentials over different instances of relational atoms). The algorithms eliminate variables, while maintaining the same form, the product of RNs.

Elimination of a relational atom X in $\phi_{RN}(X, Y)$

The first problem is to marginalize a relational atom (X) in the product of RNs with two relational atoms (X, Y): $\phi_{RN}(X, Y)$. The potential is the product of $|X|\cdot|Y|$ RNs. Note that each random variable in X has a relation with each variable in Y.

It marginalizes x_i in X, and converts the marginal into a pairwise form.

$$\int_{x_i} \prod_{y \in Y} \exp\left(-\frac{(\mathbf{x_i} - \mathbf{y})^2}{2\sigma^2}\right) = \prod_{y_i, y_j \in Y, i<j\leq|Y|} \exp\left(-\frac{(y_i - y_j)^2}{2\sigma^2 \cdot |Y|}\right) \tag{2.4}$$

Note that the marginal over $x_i \in X$ and the marginal over $x_j \in X$ ($i \neq j$) are identical. Thus, the following result is derived when it marginalizes all variables in X.

$$\begin{aligned}
&\int_{x_1} \cdots \int_{x_{|X|}} \prod_{x_i \in X} \prod_{y \in Y} \exp\left(-\frac{(\mathbf{x_i} - \mathbf{y})^2}{2\sigma^2}\right) \\
&= \prod_{x_i \in X} \left(\int_{x_i} \prod_{y \in Y} \exp\left(-\frac{(\mathbf{x_i} - \mathbf{y})^2}{2\sigma^2}\right) \right) = \left(\prod_{y_i, y_j \in Y, i<j\leq|Y|} \exp\left(-\frac{(y_i - y_j)^2}{2\sigma^2 \cdot |Y|}\right) \right)^{|X|} \\
&= \prod_{y_i, y_j \in Y, i<j\leq|Y|} \exp\left(-\frac{|X|(y_i - y_j)^2}{2\sigma^2 |Y|}\right)
\end{aligned} \tag{2.5}$$

The result of integration is the product of pairwise RNs ($\phi_{RN}(Y, Y)$) with the parameter $\frac{|X|}{2\sigma^2 \cdot |Y|}$.

Theorem 2 *For the product of RNs between two relational atoms ($\phi_{RN}(X, Y)$), 'Pairwise Constant$_1$' eliminates all ground variables of a relational atom in a*

constant time.

Proof Eliminating a variables x_i in X takes a constant time shown as Equation 2.4. Eliminating other variables in X takes a constant time shown as Equation 2.5. Thus, the computation takes only a constant time without an iteration. ■

Elimination of n random variables in $\phi_{RN}(X, X)$

The second problem is to marginalize some (n) variables in a relational atom (X) in the product of RNs within the relational atom: $\phi_{RN}(X, X)$. The potential is the product of $\frac{|X|\cdot(|X|-1)}{2}$ pairwise RNs between two ground random variables in X.

It updates the marginal after eliminating a random variable without an iteration. When it eliminate x_m, it calculates the parameters of ϕ''_{RN} given ϕ_{RN} as the following equation.

$$\int_{x_m} \prod_{1\le i<j\le m} \phi_{RN}(x_i, x_j) = \prod_{1\le i<j\le m-1} \phi_{RN}(x_i, x_j) \cdot \int_{x_m} \prod_{1\le i\le m-1} \exp\left(-\frac{(x_i - x_m)^2}{2\sigma^2}\right)$$
$$= \prod_{1\le i<j\le m-1} \phi_{RN}(x_i, x_j) \cdot \prod_{1\le i<j\le m-1} \exp\left(-\frac{(x_i - x_j)^2}{2\sigma^2 \cdot (m-1)}\right)$$
$$= \prod_{1\le i<j\le m-1} \phi_{RN}(x_i, x_j) \cdot \prod_{1\le i<j\le m-1} \phi'_{RN}(x_i, x_j) = \prod_{1\le i<j\le m-1} \phi''_{RN}(x_i, x_j)$$

The coefficient of ϕ''_{RN} is the sum of coefficient of ϕ_{RN} (σ^2) and coefficient of ϕ'_{RN} ($\sigma^2(m-1)$). The sum of two coefficients results in $\sigma^2 \cdot \frac{m-1}{m}$. Similarly, eliminating the next random variable α_{m-1} results in $\sigma^2 \frac{m-2}{m}$ (=$\sigma^2 \frac{m-1}{m} \frac{m-2}{m-1}$). Thus, eliminating n random variables results in $\sigma^2 \frac{m-n}{m}$ without iterations.

Theorem 3 *For the product of RNs with a relational atom ($\phi_{RN}(X, X)$), 'Pairwise Constant$_2$' eliminates n ground variables of the relational atom in a constant time.*

Proof Updating the parameter of $\phi_{RN}(X, X)$ from σ^2 to $\sigma^2 \frac{m-n}{m}$ takes only a constant time. ■

2.4.3 A Linear Time Relational Atom Elimination

This section provides a linear time variable elimination algorithm $O(|U|)$ which can be applied to any product of RNs. This algorithm is used when the constant time algorithms of the previous sections are not applicable.

Elimination of multiple atoms in $\prod \phi_{RN}(X_i, X_j)$

This problem is to marginalize some variables in U, ($U = \{X_1, X_2, \cdots, X_{|N|}\}$) in the product of RNs between two relational atoms: $\prod \phi_{RN}(X_i, X_j)$. If all relational atoms have pairwise relationships among each other, there are $\frac{|N| \cdot |N-1|}{2}$ pairwise RNs.

Lemma 4 *For $|U|$ variables in $|N|$ relational atoms ($U = \{X_1, X_2, \cdots, X_{|N|}\}$) and RN potentials, marginalizing n variables in a ground model takes $O(n \cdot |U|^2)$.*

Proof Suppose we eliminate a variable $x \in U$. Eliminating a variable x in RN needs updates coefficients of terms $(x_i x_j)$ where x_i and x_j have relations with the variable x. When x has relations with all other variables in U, the number of terms is bounded by $O(|U|^2)$. Thus, eliminating n variables takes $O(n \cdot |U|^2)$ because it needs n iterations. ■

Thus, any inference algorithm in a ground model has an order of $O(|U|^3)$ time complexity, when it eliminates all ground variables except a few query variables.

To reduce the time complexity, our lifted algorithm uses following notations which refer ground variables in an atom X compactly: $X_{[m]} = \sum_{1 \le i \le m} x_i$; $X_{[m]^2} = \sum_{1 \le i \le m} x_i^2$; and $X_{[m][m]} = \sum_{1 \le i < j \le m} x_i \cdot x_j$. The notations give the following properties (when $|X| = m$ and $|Y| = n$):

$$(X_{[m]})^2 = X_{[m]^2} + 2X_{[m][m]}$$

$$\begin{aligned} exp\left(2X_{[m][m]} - (m-1)X_{[m]^2}\right) &= \prod_{x_i, x_j \in X} exp\left(-(x_i - x_j)^2\right) = \phi'_{RN}(X, X) \\ exp\left(2X_{[m]}Y_{[n]} - nX_{[m]^2} - mY_{[n]^2}\right) &= \prod_{x_i \in X, y_k \in Y} exp\left(-(x_i - y_k)^2\right) = \phi''_{RN}(X, Y) \end{aligned}$$

For the product of potentials over X, Y, and $\{x'\}$, our algorithm marginalizes x':

$$\begin{aligned} &\int_{x'} \phi_{RN}(X, x') \cdot \phi_{RN}(Y, x') \\ &= \int_{x'} exp\left(-(m+n)x'^2 + 2(X_{[m]} + Y_{[n]})x' - (X_{[m]^2} + Y_{[n]^2})\right) \\ &= \sqrt{\frac{\pi}{m+n}} \cdot exp\left(\frac{(X_{[m]} + Y_{[n]})^2}{m+n} - (X_{[m]^2} + Y_{[n]^2})\right) \\ &= c \cdot exp\left(\frac{2X_{[m][m]} + 2X_{[m]}B_{[n]} + 2Y_{[n][n]} - (m+n-1)(X_{[m]^2} + Y_{[n]^2})}{m+n}\right) \\ &= c \cdot \phi'_{RN}(X, X) \cdot \phi''_{RN}(X, Y) \cdot \phi'''_{RN}(Y, Y) \end{aligned} \tag{2.6}$$

It iterates until all n variables are eliminated.

Theorem 5 *For $|U|$ variables in $|N|$ relational atoms ($U = \{X_1, X_2, \cdots, X_{|N|}\}$) and potentials in RN, 'Pairwise Linear' eliminates n variables in $O(n \cdot |N|^2)$.*

Proof WLOG, we marginalize a variable $x' \in X_1$. We make an artificial atom Y which includes all relational atoms, when those atoms have relationships with X_1.[5] Then, $\{x'\}$ is split from X_1 ($X_1 = X'_1 \cup \{x'\}$ and $X'_1 \cap \{x'\} = \emptyset$). When

[5]That is, $Y = \bigcup_i X'_i$ and $X'_i = \{\frac{x}{\sigma_i} | x \in X_i\}$, when σ_i is the variance used in $\phi_{RN}(X_1, X_i)$.

marginalizing x' out in $\phi_{RN}(X'_1, x') \cdot \phi_{RN}(Y, x')$, the marginal is also the product of RNs shown as Equation 2.6: $\phi'_{RN}(X'_1, X'_1) \cdot \phi''_{RN}(X'_1, Y) \cdot \phi'''_{RN}(Y, Y)$.

The marginal can be represented without the artificial atom Y in the following procedures. We convert into $\phi''_{RN}(X', Y)$ and $\phi'''_{RN}(Y, Y)$ as follows. First, $\phi''_{RN}(X'_1, Y)$ is represented as the product of *RNs* between atoms X_i in Y and X'_1: $\prod_{X_i \in Y} \phi''_{RN}(X'_1, X_i)$. Second, $\phi'''_{RN}(Y, Y)$ is also represented as the product of *RNs* between atoms X_i and X_j in Y: $\prod_{X_i, X_j \in Y} \phi''_{RN}(X_i, X_j)$.

For each elimination, it updates parameters of all possible pairs $O(|N|^2)$ among $|N|$ atoms. Thus, the computational complexity to eliminate n variables is the order of $O(n \cdot |N|^2)$. ■

Thus, 'Pairwise *Linear*' has linear time complexity $O(|U|)$ with respect to the number of ground variables.

2.5 Exact Lifted Inference with RCMs

This section presents our algorithm, *ELIMINATE-CONTINUOUS*, which generates a new parfactor after eliminating a set of relational atoms given a set of parfactors. A potential of each parfactor is the product of *Relational Pairwise Potentials (RPPs)*:

$$\phi_{RPP}(X, Y) = \prod_{x \in X, y \in Y} \phi_{RPP}(x, y)$$

A *relational pairwise model* is a RCM whose potentials are RPPs. Here, *RPPs* are not limited to the RNs in Section 2.4.1.

2.5.1 Conditions for Exact Lifted Inference

The lifted *ELIMINATE CONTINUOUS* algorithm provides the exact solution for potentials of parfactors when the potentials satisfy three conditions: *Condition*

(I), analytically integrable; *Condition (II)*, closed under product operations; and *Condition (III)*, closed under marginalizations, thus represented with the product of *relational pairwise potentials* again. The RNs are an example that satisfies the conditions. Here, we introduce another potential, a linear Gaussian, which satisfies the conditions.

Lemma 6 *The product of RNs with non-zero Means (RNMs) satisfies the three conditions. A RNM has the following form (d is a constant).*

$$\phi_{RN}(X,Y) = \prod_{x\in X, y\in Y} \frac{1}{\sigma\sqrt{2\pi}} exp\left(-\frac{(x-y-d)^2}{2\sigma^2}\right)$$

The proof is provided in Section 2.9.

2.5.2 Inversion-Elimination

Inversion elimination is applicable when the set of logical variables in g is same with the set of logical variables in e, $LV(e) = LV(g)$. Let θ_1,...,θ_n be enumeration of Θ_g.

$$\begin{aligned}
\int_{RV(e)} \phi(g) &= \int_{RV(e)} \prod_{\theta\in\Theta_g} \phi_g(A_g\theta) = \int_{e[\theta_1]} \cdots \int_{e[\theta_n]} \phi_g(A_g\theta_1)\cdots\phi_g(A_g\theta_n) \\
&= \prod_{\theta\in\Theta_g} \int_{e[\theta]} \phi_g(A_g\theta) (\because \textit{split} \text{ (Section 2.3)}) = \prod_{\theta\in\Theta_g} \int_e \phi_g(A'\theta, e) \\
&= \prod_{\theta\in\Theta_g} \phi'(A'\theta) = \phi_{g'}
\end{aligned}$$

Return to the econometric market example, inversion elimination can be applied to $G[S,B]$. Before an elimination, it combines two parfactors which include ϕ_2 and ϕ_3 respectively. The combined parfactor is $g = (\{S,B\}, \top, (M[S], G[S,B], R[B]), \phi_2 \cdot$

ϕ_3). Then, the elimination procedure is follow.

$$\begin{aligned}
\int_{RV(G)} \phi(g) &= \int_{RV(G)} \prod_{s\in S, b\in B} \phi_g(M(s), G(s,b), R(b)) \\
&= \prod_{s\in\{auto,\cdots,stock\}, b\in\{b_1,\cdots,b_m\}} \left(\int_{G(s,b)} \phi_g(M(s), G(s,b), R(b)) \right) \\
&= \prod_{s\in\{auto,\cdots,stock\}, b\in\{b_1,\cdots,b_m\}} \phi_{new}(M(s), R(b)) = \phi_{new}(M[S], R[b]) = \phi_{g'}
\end{aligned}$$

Note that, the number of substitutions ($|\Theta_g|$) is the number of market sectors ($|S|$) times the number of banks ($|B|$). Regardless the number of substitutions, we can apply the same integration to eliminate $|S| \cdot |B|$ number of random variables (G(s,b)). Thus, it calculates the integral ($= \int_L \phi_g(M(s), G(s,b), R(b))$) one time regardless of specific s and b. The marginal ($\phi_{new}(M[S], R[B])$) becomes the potential of the output parfactor (g').

2.5.3 Relational-Atom-Elimination

Relational-Atom-Elimination marginalizes atoms when *Inversion-Elimination* is not applicable. It is a generalized algorithm of those for *RN* shown in Section 2.4. It marginalizes each relational atom of a parfactor g according to three cases: (1) variables in the atom e has relationship with an atom (i.e. '$\phi(X, Y)$'); (2) variables in the atom e has relationships only each other (i.e. '$\phi(X, X)$'); and (3) other general cases (i.e. '$\prod \phi(X_i, X_j)$').

For the case (1), a modified '*Pairwise Constant*$_1$' eliminates an atom e. In this case, integrating out a random variable in the atom does not affect integrating another variable in the atom as shown in Section 2.4.2. That is, $\int_{RV(e)} \prod_{\theta\in\Theta_g} \phi_g(.) = \prod_{\theta_e\in\Theta_e} \int_{e(\theta_e)} \prod_{\theta\in\Theta_{g\setminus\{e\}}} \phi_g(.)$. Here, E is the set of atoms in g, and $\overline{E} = E \setminus \{e\}$, and

Θ_E is the set of all substitutions for E.

$$\begin{aligned}
\int_{RV(e)} \phi(g) &= \int_{RV(e)} \prod_{\theta \in \Theta_E} \phi_g(A_g\theta) = \int_{RV(e)} \prod_{\theta_e \in \Theta_{\{e\}}} \prod_{\theta \in \Theta_{E\setminus\{e\}}} \phi_g(A_g\theta_e, A_g\theta) \\
&= \prod_{\theta_e \in \Theta_{\{e\}}} \int_{e[\theta_e]} \prod_{\theta \in \Theta_{E\setminus\{e\}}} \phi_g(A_g\theta_e, A_g\theta) = \prod_{\theta_e \in \Theta_{\{e\}}} \phi'(RV(\overline{E}))(\because \textit{Condition}(I)) \\
&= \phi'(RV(\overline{E}))^{|RV(e)|} = \phi''(RV(\overline{E}))(\because \textit{Condition}(II))
\end{aligned}$$

Normally, the marginal $\phi''(RV(\overline{E}))$ is not a *relational pairwise potential* because all random variables in $\overline{E}$ are arguments of the potential. However, when *Condition (III)* is satisfied, the marginal can be converted into the product of *relational pairwise potentials*: $\phi''(RV(\overline{E})) = \prod_{X_i, X_j \in RV(\overline{E})} \phi_{RPP}(X_i, X_j)$.

In the financial example, it eliminates $R[B]$ as follow.

$$\begin{aligned}
\int_{RV(R)} \phi(g') &= \int_{RV(R)} \prod_{s \in S, b \in B} \phi_{new}(M(s), R(b)) \\
&= \prod_{b \in B} \int_{R(b)} \prod_{s \in S} \phi_{new}(M(s), R(b)) = \prod_{b \in B} \phi'_{new}(M(auto), \cdots, M(stock)) \\
&= \phi'_{new}(M(auto), \cdots, M(stock))^{|RV(R)|} = \phi''_{new}(M(auto), \cdots, M(stock))
\end{aligned}$$

Beyond Relational Gaussian defined in Section 2.4.1, any potential function satisfying the Condition III) can convert the potential ϕ''_{new} into the pairwise form $\prod \phi'''_{new}$.

$$\phi''_{new}(M(auto), \cdots, M(stock)) = \prod_{s_1, s_2 \in S} \phi'''_{new}(M(s_1), M(s_2))$$

Likewise, for the cases (2) and (3), generalized algorithms of '*Pairwise Constant*$_2$' and '*Pairwise Linear*' are also applied respectively.

2.6 Related Work

[Poole, 2003] solves inference problems with the unification which dynamically splits a set of ground nodes and unifies them. With a counting formula, [de Salvo Braz *et al.*, 2005; 2006] provides a tractable algorithm. [Milch *et al.*, 2008] applies the counting formula to reduce the size of probability density tables. However, these lifted inference algorithms are hard to apply to continuous domains.

MLNs (Markov Logic Network) [Richardson and Domingos, 2006] use First-order logic sentences to represent relationships over nodes in a graphical model. In this regard, MLNs also represent graphical models at the relational level. [Singla and Domingos, 2008] provides an approximated lifted inference algorithm over discrete domain. [Singla and Domingos, 2007] makes an analysis for infinitely many discrete variables. However, these achievements are not for continuous domains, too. Although there is an inference algorithm for Hybrid MLNs [Wang and Domingos, 2008], it is an approximated algorithm. Thus, most of achievements are comparable to lifted inferences [de Salvo Braz *et al.*, 2005; Milch *et al.*, 2008; Pfeffer *et al.*, 1999] over discrete domain.

Inference with Gaussian distributions is a classic problem [Roweis and Ghahramani, 1999]. In detail, calculating conditional densities of multivariate Gaussians requires matrix inversions [Kotz *et al.*, 2000] which are intractable for high dimensions. [Lerner and Parr, 2001; Shenoy, 2006] builds inference algorithms for hybrid models with Gaussians. [Paskin, 2003] shows that efficient inference is possible for a linear Gaussian when the treewidth of the model is small. For models with large treewidth, however, those inference algorithms over ground models which would be inefficient.

Recent advances in inference with relational models [Kisynski and Poole, 2009; Mihalkova and Mooney, 2009] show the promise of the approach in discrete models, and underline the promise of our algorithm in continuous models.

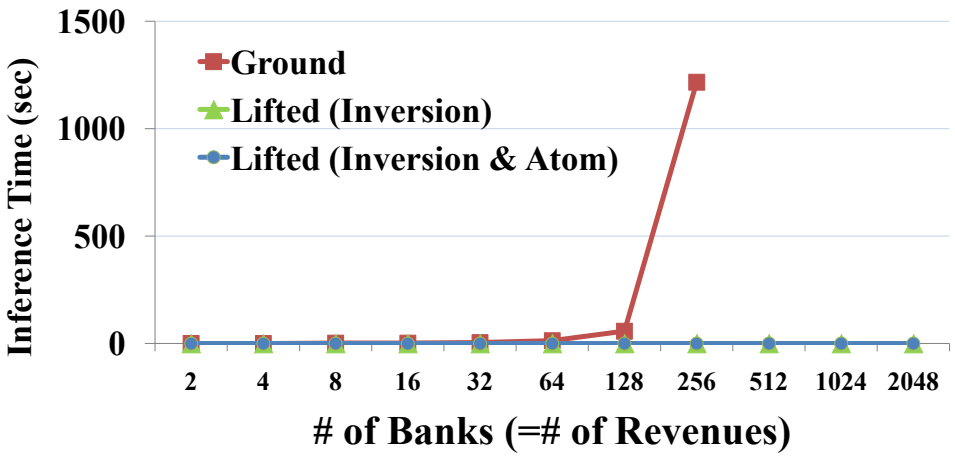


Figure 2.5: Inference time with different number of banks

2.7 Experimental Results

We report experiments for the recession model provided in the chapter. For experiments, we implemented three algorithms: (A) inference with a grounded model; (B) inference with only Inversion-Elimination; and (C) inference with both Inversion-Elimination and Relational-Atom-Elimination. Our new algorithm (C) is significantly faster than the grounded model (A) and Inversion-Elimination (B). Note that Inversion-Elimination (B) is also our new algorithm for continuous variables, even though comparable elimination methods for discrete variables [de Salvo Braz *et al.*, 2005; Milch *et al.*, 2008; Pfeffer *et al.*, 1999] existed prior to ours. Our experimental results are shown in Figure 2.5 and 2.6

In the recession model, we provided observations for one market variable (M) and one revenue variable (R).[6] Those variables were split from relational atoms. Then, we calculated the marginal density of the Recession variable. We increased the number of markets and the number of banks from 2 to 2048 exponentially. We set an hour of cut-off time. With 512 banks, the grounded inference (A) did not complete within an hour. Meanwhile, the Inversion Elimination (B) and our new algorithm (C) finished computations in almost a constant time even for 2048

[6]Observations are required to make the product of RNs a probability density function. Please refer Lemma 1 for details.

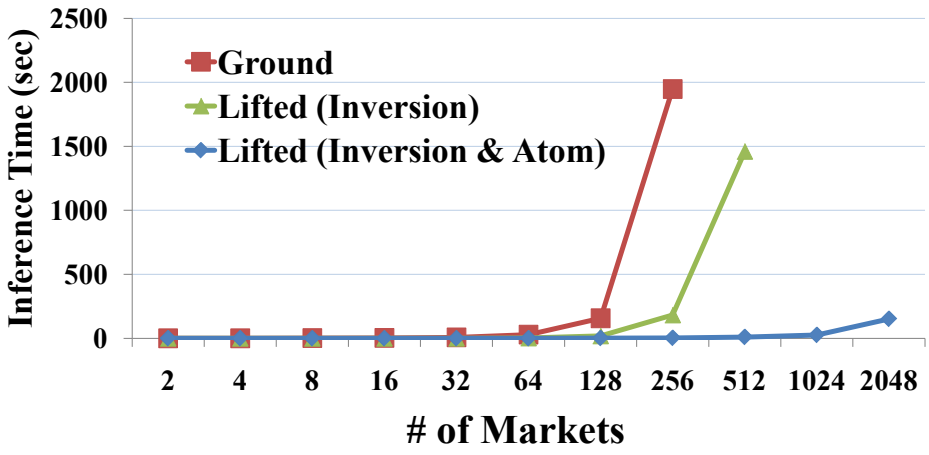


Figure 2.6: Inference time with different number of markets

banks. With 512 markets, (A) could not finish within an hour, again. With 1024 markets, (B) did not finish in an hour. Meanwhile, our new algorithm (C) finished in a reasonable time (about 151 secs) even with 2048 markets.

2.8 Conclusion and Future work

In this chapter, we propose a new exact lifted inference algorithm for Relational Continuous Models (RCMs). This algorithm is an advancement of exact inference in RCMs, since all previous works are restricted to discrete domain. Given a query and observations, our algorithm exactly computes the conditional density of the query, when potentials satisfy specified conditions.

There are two limitations in our current algorithm. First, found potentials which satisfy the conditions in Section 2.5 are variants of Gaussian potentials. Thus, finding potentials beyond Gaussian is a goal of our future works. Second, the current algorithm is designed only for continuous variables. Many real-world models require not only continuous variables but also discrete variables. Thus, making an efficient inference algorithm for hybrid relational models would be a promising direction.

2.9 Appendix

Proof of Lemma 1 Here, we prove that the product of RNs integrates to a constant given the conditions. The constant becomes the normalizing factor of the probability density function.

We prove this by contradiction. Suppose that the product of RNs does not integrate to a constant. That is, it integrates to infinity.

According to Equation 2.2, the product of RNs maintains the same form after integrating out a random variable x. Thus, only possible case to be infinity is when the marginal (after an integration over x) is a constant function of another random variable y which is not yet integrated.

When x has relations with more than one variable (e.g. y and z), the condition for infinity is not satisfied. The marginal includes a potential $\phi(y, z)$. When x has a relation with only y which has relations with other variables beyond x, the condition for infinity is not satisfied. The marginal is not a constant function of y.

Thus, only $\phi(x, y)$ satisfies the condtion for infinity. Given the assumption that at least a RN includes a constant, y can not be a variable. Thus, it contradicts the assumption. ■

Proof of Lemma 6 First, the product of RNMs is analytically integrated by the rule in Equation 2.3. Thus, the product of RNMs satisfy the *Condition (I)*.

Second, the product of RNMs is closed under product operations and marginalizations. It satisfies the *Condition (I)* because it is an exponential family. That is, the product of two RNMs ($\phi'_{RNM}(x, y)$ and $\phi''_{RNM}(x, y)$) is another RNMs (ϕ'''_{RNM} (x, y)). Thus, the product of RNMs satisfies the *Condition (II)*.

Third, it is also closed under marginalizations. When y_j in Equation 2.4 is

substituted with $y_j - d$, the following equation is derived.

$$\int_{x_i} \prod_{y \in Y} \exp\left(-\frac{(\mathbf{x_i} - \mathbf{y} - \mathbf{d})^2}{2\sigma^2}\right) = \prod_{y_i, y_j \in Y} \exp\left(-\frac{(y_i - y_j - 0))^2}{2\sigma^2 \cdot |Y|}\right)$$

Thus, the result is the product of RNMs.

As explained in the proof of Theorem 5, the the product of RNMs can be represented as the following form $\phi_{RNM}(X, x') \cdot \phi_{RNM}(Y, x')$ when x' is the variable of integration.

When $y_j \in Y$ in Equation 2.6 is substituted with $y_j - d \in Y'$, the following equation is derived.

$$\begin{aligned}
\int_{x'} \phi_{RNM}(X, x') \cdot \phi_{RNM}(Y, x') &= c \cdot \int_{x'} \phi_{RN}(X, x') \cdot \phi_{RN}(Y', x') \\
&= c' \cdot \phi'_{RN}(X, X) \cdot \phi''_{RN}(X, Y') \cdot \phi'''_{RN}(Y', Y') \\
&= c'' \cdot \phi'_{RNM}(X, X) \cdot \phi''_{RNM}(X, Y) \cdot \phi'''_{RNM}(Y, Y)
\end{aligned}$$

The result is also the product of RNMs. Thus, it is closed under marginalizations.

■

CHAPTER 3

LIFTED RELATIONAL KALMAN FILTERING

Kalman filtering is a computational tool with widespread applications in robotics, financial and weather forecasting, environmental engineering and defense. Given observation and state transition models, the Kalman filter (KF) recursively estimates the state variables of a dynamic system. However, the KF requires a cubic time matrix inversion operation at every timestep which prevents its application in domains with large numbers of state variables. We propose Relational Gaussian Models to represent and model dynamic systems with large numbers of variables efficiently. Furthermore, we devise an exact lifted Kalman filtering algorithm which takes only linear time in the number of random variables at every timestep. We prove that our algorithm takes linear time in the number of state variables even when individual observations apply to each variable. To our knowledge, this is the first lifted (linear time) algorithm for filtering with continuous dynamic relational models.

3.1 Introduction

Many real-world systems can be modeled by continuous variables and relationships (or dependences) among them. The Kalman filter (KF) [Kalman, 1960] accurately estimates the state of a dynamic system given a sequence of control-inputs and observations. It has been applied in a broad range of domains which include weather forecasting [Burgers *et al.*, 1998], localization and tracking in robotics [Limketkai *et al.*, 2005], economic forecasting in finance [Bahmani-Oskooee and Brown, 2004] and many others. Given a sequence of observations

and Gaussian dependences between variables, the filtering problem is to calculate the conditional probability density of the state variables at each timestep. Unfortunately, the KF computations are cubic in the number of random variables which limits current exact methods to domains with limited number of random variables. This has led to the combination of approximation and sampling (e.g. the Ensemble Kalman filter [Evensen, 1994]).

This chapter leverages the ability of relational languages [Friedman *et al.*, 1999; Poole, 2003; Milch *et al.*, 2005; Richardson and Domingos, 2006] to specify models with size of representation independent of the size of populations involved. Various lifted inference algorithms for relational models have been proposed [Poole, 2003; de Salvo Braz *et al.*, 2005; Milch and Russell, 2006; Singla and Domingos, 2008; Wang and Domingos, 2008; Choi *et al.*, 2010a]. These seek to carry computations in time independent of the size of the populations involved. However, the key challenge in relational filtering (of dynamic systems) is ensuring that the representation does not degenerate to the ground case when multiple observation are made. As more observations are received, an increasing number of objects become distinguished. This precludes the application of previously known algorithms unless approximately equivalent objects are grouped with expensive clustering algorithms.

We propose Relational Gaussian Models (RGMs) to model dynamic systems of large number of variables in a relational fashion. RGMs have as their main building block the pairwise linear Gaussian potential as detailed in Section 3.2. Further, we propose a new lifted filtering algorithm that is able to marginalize out random variables of the previous timestep efficiently (in time linear in the number of random variables) while maintaining the relational (RGM) representation. This prevents the model from being increasingly grounded even when individual observations are made for all random variables. Moreover, updating the relational representation takes only quadratic time in the number of relational atoms (sets of random variables). One key insight is that, given identical observation models, even when the means of the random variables are dispersed their variances remain

identical. This is sufficient to maintain a relational representation.

This chapter is organized as follows. Section 3.2 introduces definitions and the relational filtering problem. Section 3.3 presents our main technical results, i.e., the recursive estimation of the states of random variables in a lifted fashion. Section 3.4 presents our algorithm in detail together with complexity results. Section 3.6 shows experimental results with a housing market model. Section 3.5 discusses previous work. The chapter concludes in Section 3.7.

3.2 Model and Problem Definitions

In this section, we define Relational Gaussian Models (RGMs) and introduce the filtering problem for dynamic relational models.

3.2.1 Relational Continuous Models

Dependencies between variables are represented using **Parfactor** models[1], i.e. parameterized factor models. Each *parfactor* $g = (L, C, \mathbf{X}, \phi)$ is composed of a set of logical variables (or objects) (L), constraints on L (C), a list of relational atoms ($\mathbf{X}$), and a potential function on $\mathbf{X}$ (ϕ).

Relational atoms represent the set of random variables corresponding to all ground substitutions of its logical variables. Formally, applying a substitution θ to a parfactor g yields a new parfactor $g\theta = (L', C\theta, \mathbf{X}\theta, \phi)$, where L' is obtained by renaming the variables in L according to θ. If θ is a ground substitution, $g\theta$ is a factor. A factor $f = (\mathbf{x}, \phi)$ is a pair where $\mathbf{x}$ is a list of ground random variables $(x_1, \ldots, x_{|\mathbf{x}|})$ and ϕ is a potential on $\mathbf{x}$, a function from $range(\mathbf{x}) = \times_{i=1}^{|\mathbf{x}|} range(x_i)$ to $\mathbb{R}^+$. A factor f defines a weighting function on a valuation $v = (v_1, \ldots, v_{|\mathbb{X}|})$: $w_f(v) = \phi(v_1, \ldots, v_{|\mathbf{x}|})$. The weighting function for a *parfactor* g is the product of the weighting functions of all of its ground substitutions (factors), $w_g(v) =$

[1] Our representation is based on previous work [Poole, 2003; de Salvo Braz *et al.*, 2005; Milch and Russell, 2006; Choi *et al.*, 2010a].

$\prod_{f \in g} w_f(v)$. Hence, a set of parfactors G defines[2] a probability density proportional to,

$$w_G(v) = \prod_{g \in G} \prod_{f \in g} w_f(v). \tag{3.1}$$

3.2.2 Relational Gaussian Models

Relational Gaussian Models (RGMs) are a subset of Relational Continuous Models (RCMs) where potentials are restricted to be Gaussian distributions. RGMs are composed of three types of parfactor models: (1) Relational Transition Models (RTMs); (2) Relational Pairwise Models (RPMs); and (3) Relational Observation Models (ROMs). Suppose that we have n relational atoms: $X_t^1(L)$, ..., $X_t^n(L)$ where L is a list of logical variables. In a relational linear dynamic model, relational atoms are linearly influenced by control-inputs $U_t^1(L)$, ..., $U_t^n(L)$. Similarly, a linear observation model specifies the relationship between observation variables $O_t^1(L)$, ..., $O_t^n(L)$ and other relational atoms. Control inputs and observations are associated with relational atoms in two ways: (1) direct association; and (2) indirect association. We provide further details in Section 3.2.4.

Relational Transition Models (RTMs) model the dependence of relational atoms at the next timestep, $X_{t+1}^j(a')$, on relational atoms at the current timestep, $X_t^i(a)$, and (when available) control-input information. They take the following form,

$$X_{t+1}^j(a') = B_X^{i,j} \cdot X_t^i(a) + B_U^{i,j} \cdot U_t^i(a) + G_{RTM}^{i,j}, \tag{3.2}$$

where $G_{RTM}^{i,j} \sim N(0, \sigma_{RTM}^{i,j})$ and $N(m, \sigma^2)$ is the normal distribution with mean m and variance σ^2. $B_X^{i,j}$ and $B_U^{i,j}$ are the transition models, matrices or a constants, corresponding to two relational atoms.

For univariate state variables, we can represent the transition model with a lin-

[2]The condition is that at least a random variable has a prior distribution as outlined in [Choi *et al.*, 2010a].

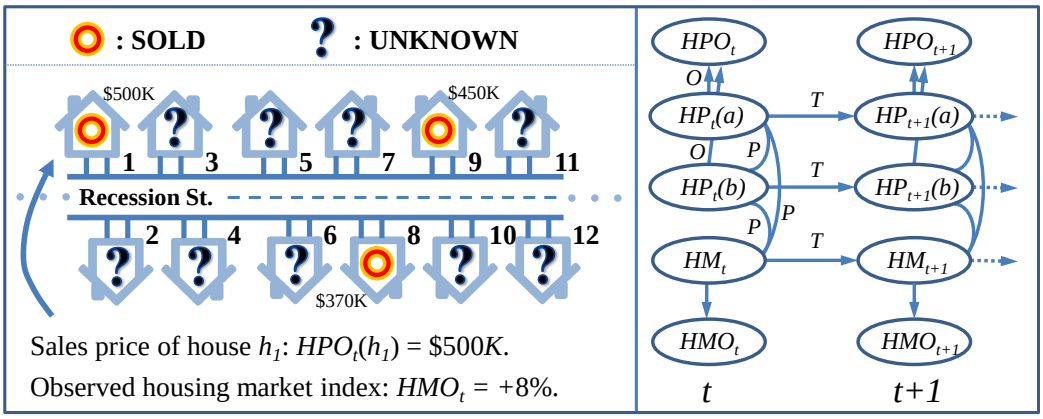


Figure 3.1: Example of a housing market model. We are interested in estimating the hidden value of houses given observations of house sales prices (e.g. $HPO_t(1)$ = \$500K). Both, the hidden value of a house and the observed sales prices are affected by several factors, e.g., house values increase by a certain rate every year and are also influenced by a housing market index (HM_t).

ear Gaussian,

$$\phi_{RTM}(X^j_{t+1}(a')|X^i_t(a), U^i_t(a)) \propto \exp\left(-\frac{(X^j_{t+1}(a') - B^{ij}_X \cdot X^i_t(a) - B^{ij}_U \cdot U^i_t(a))^2}{2 \cdot {\sigma^{ij}_{RTM}}^2}\right). \tag{3.3}$$

The most common transition is the transition from the current state $X^i_t(a)$ to the next $X^i_{t+1}(a)$. It is represented as follows,

$$X^i_{t+1}(a) = B^i_X \cdot X^i_t(a) + B^i_U \cdot U^i_t(a) + G^i_{RTM}. \tag{3.4}$$

Relational Observation Models (ROMs) represent the relationships between the hidden (state) variables, $X^i_t(a)$, and the observations made at the corresponding timestep, $O^i_t(a)$,

$$O^i_t(a) = H^i_t \cdot X^i_t(a) + G^i_{ROM}, \tag{3.5}$$

where $G^i_{ROM} \sim N(0, \sigma^i_{ROM})$. H^i_t is the observation model, a matrix or a constant, between the hidden variables and the observations.

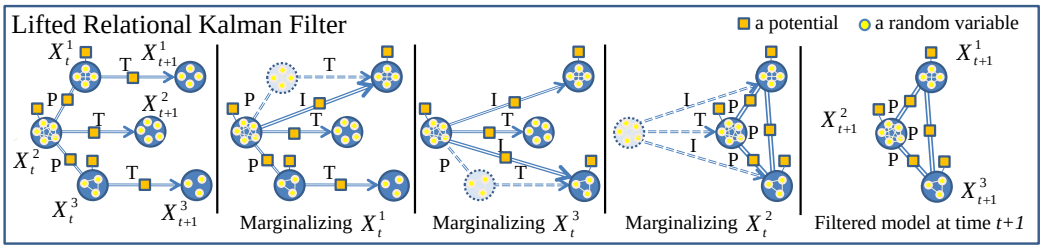


Figure 3.2: This model has three relational atoms, X_i, which may *represent* any number of random variables. The relational representation dramatically eliminates the need for redundant potentials. Hence, representation and filtering become much more efficient than in the propositional case. Note that the conventional KF representation is not suited for efficient (i.e. lifted) inference.

In the linear Gaussian representation, they take the following form,

$$\phi_{ROM}(O_t^i(a)|X_t^i(a)) \propto \exp\left(-\frac{(O_t^i(a) - H_t^i \cdot X_t^i(a))^2}{2 \cdot {\sigma_{ROM}^i}^2}\right). \tag{3.6}$$

Relational Pairwise Models (RPMs) represent Gaussian dependences between pairs of relational atoms within the same timestep as follows,

$$X_t^i(a) = R_t^{i,j} \cdot X_t^j(a') + G_{RPM}^{i,j}, \tag{3.7}$$

where $G_{RPM}^{i,j} \sim N(0, \sigma_{RPM}^{i,j})$. $R_t^{i,j}$ is the pairwise coefficient, a matrix or a constant, between the two relational atoms.

Note that RTMs and ROMs are directed models while RPMs are undirected. The directed models represent the nature of dynamic systems (e.g. the state at the next timestep depends on the current timestep). The product of RPMs is an efficient way to represent a multivariate Gaussian density over all the state variables. [3]

[3] Note that a multivariate Gaussian density (of state variables) is a quadratic exponential form. The quadratic exponential form can always be decomposed into terms involving only single variables and pairs of variables. We provide additional details in Section 3.5.

3.2.3 A Relational Filtering Problem

Given a prior (or current belief) over the state variables, the filtering problem is to compute the posterior after a sequence of timesteps. The input to the problem is: (1) Relational Gaussian Model (RTMs, RPMs and ROMs); (2) current belief over the relational atoms (X_0^i) represented by a product of relational Gaussian potentials; (3) sequence of control-inputs ($U_1^i, \ldots, U_T^i$); and (4) sequence of observations ($O_1^i, \ldots, O_T^i$). The output is the relational Gaussian posterior distribution over the relational atoms (X_T^i) at timestep T.

3.2.4 Input and Observation Association

At every timestep the control-inputs and observations must be associated with the random variables they affect. The ideas in this section apply to control-inputs and observations but we illustrate them for observations.

We distinguish two types of observations: direct and indirect. Direct observations are those made for a specific random variable. For instance, if we make an observation for each random variable in a subset $A_t^i \subseteq X_t^i$ of the ground substitutions of relational atom X_t^i, we are looking at the following model,

$$\prod_{a_j \in A_t^i} \phi_{ROM}\left(o_t^i(a_j) | X_t^i(a_j)\right). \tag{3.8}$$

In the example of Figure 3.1, observing the selling price of a house would dramatically reduce the variance of the hidden variable that represents the true value of that house.

Similarly, multiple direct observations, $O_t^i = o_t^{i,1}, o_t^{i,2}, \ldots, o_t^{i,|O_t^i|}$, could be made for each variable in some set of random variables,

$$\prod_{a_j \in A_t^i} \prod_{o_t^{i,k} \in O_t^i} \phi_{ROM_k}\left(o_t^{i,k}(a_j) | X_t^i(a_j)\right). \tag{3.9}$$

Given some notion of neighborhood (e.g. a residential neighborhood or a block

of houses), indirect observation allows the possibility that observations made for a random variable, $o_t^i(a')$, would influence nearby random variables, $X_t^i(a_j)$, $a' \neq a_j$,

$$\prod_{a_j \in A_t^i} \phi_{ROM}\left(o_t^i(a') | X_t^i(a_j)\right). \tag{3.10}$$

For example, this allows the possibility that the observation of the selling price of a house would reduce the variance of the true values of neighboring houses.

Current (exact) lifted inference algorithms (e.g. [Kersting *et al.*, 2006; Choi *et al.*, 2010a]) handle observations by partitioning the relational atoms into groupings of groundings for which identical observations and observation models apply. In contrast, our approach partitions a relational atom into sets according to the number of different types of observations associated with each random variable. For instance, if an individual observation of the same ROM type is made for each random variable then no partitioning at all is necessary. The intuition for this is that the filtering process will assign the same variance to any two hidden variables for which the same number of observations is made at the current timestep.

Here, the partition will determine new RPMs, the pairwise parfactors which maintain the variances and covariances. In particular, the number of new RPMs is quadratic in the size of the partition. Since individual observations cause the means of the random variables to differ we store the mean information in the prior and posterior (P and P_{new} in Section 3.3). Hence, the number of priors and posteriors is linear in the number of random variables. However, this will not affect the computational complexity of inference as long as the RPMs do not degenerate. Further details are given in Sections 3.3.3 and 3.4.

Formally, given a partition $\Pi^i = (M_1^i, M_2^i, \ldots, M_{|\Pi^i|}^i)$ of a relational atom, X^i, the observation model takes the form,

$$\prod_{M_l^i \in \Pi^i} \prod_{a_j \in M_l^i} \prod_{o^{i,k} \in O_l^i} \phi_{ROM_k}\left(o^{i,k}(a_j) | X^i(a_j)\right), \tag{3.11}$$

where we omit the time subscript and where O_l^i is the set of observations relevant to part l.

3.3 Lifted Relational Kalman Filter

The **Lifted Relational Kalman filter (LRKF)**, just like the conventional Kalman filter, carries two recursive computations: prediction step and update/correction step.

3.3.1 Lifted Prediction

In the prediction step, our current belief over the states of the relational atoms together with the RTMs, RPMs and control-inputs are used to make a best estimate of state without observation information. First, the product of potentials in the RTMs and RPMs is built. Second, the variables from the previous timestep are marginalized resulting in new RPMs and estimates of the relational atoms in the current timestep. We call this estimates the *intermediate posterior*, the input to the update step.

$$\begin{aligned}
&\int_{X_t^1,\dots,X_t^n} \prod_{1\le i<j\le n} \prod_{\substack{a\in A^i\\ a'\in A^j}} \phi_{RTM}^{i,j}\left(X_{t+1}^j(a')|X_t^i(a), U_t^i(a)\right) P^i\left(X_t^i(a)\right) \phi_{RPM}^{i,j}\left(X_t^i(a), X_t^j(a')\right) \\
&= \int_{X_t^1,\dots,X_t^n} \prod_{1\le i<j\le n} \prod_{\substack{a\in A^i\\ a'\in A^j}} \exp\left(-\frac{\left(X_{t+1}^j(a') - B_X^{i,j} X_t^i(a) - B_U^{i,j} U_t^i(a)\right)^2}{{\sigma_{RTM}^i}^2}\right) \\
&\quad \cdot \exp\left(-\frac{\left(X_t^i(a) - \mu_P^i(a)\right)^2}{{\sigma_P^i}^2}\right) \cdot \exp\left(-\frac{\left(X_t^i(a) - R_t^{i,j} X_t^j(a')\right)^2}{{\sigma_{RPM}^{i,j}}^2}\right) \qquad (3.12) \\
&= \prod_{1\le i<j\le n} \prod_{\substack{a\in A^i\\ a'\in A^j}} \exp\left(-\frac{\left(X_{t+1}^i(a) - R'^{i,j}_{t+1} X_{t+1}^j(a')\right)^2}{{\sigma'^{i,j}_{RPM}}^2}\right) \cdot \exp\left(-\frac{\left(X_{t+1}^i(a) - \mu_{P'}^i(a)\right)^2}{{\sigma_{P'}^i}^2}\right) \\
&= \prod_{1\le i<j\le n} \prod_{\substack{a\in A^i\\ a'\in A^j}} \phi'^{i,j}_{RPM}\left(X_{t+1}^i(a), X_{t+1}^j(a')\right) \cdot \prod_{1\le i\le n} \prod_{a\in A^i} P'^i\left(X_{t+1}^i(a)\right). \qquad (3.13)
\end{aligned}$$

Here, ${\phi'}^{i,j}_{RPM}$, P^i and P'^i are respectively the updated RPMs, the priors and the intermediate posteriors. More details of the integration are given in Appendix 3.8.

3.3.2 Lifted Update

In the update step, the intermediate posterior P'^i and ROMs are used to correct our estimate of the relational atoms.

When a single observation, o^i_{t+1}, is associated with all variables in a relational atom, we calculate the posterior for one random variable $X^i_{t+1}(a)$ and use the result for the rest of the groundings of the same relational atom,

$$
\begin{aligned}
&P'^i(X^i_{t+1}(a)) \cdot \phi_{ROM}\left(o^i_{t+1}|X^i_{t+1}(a)\right) \\
&= \exp\left(-\frac{\left(X^i_{t+1}(a) - \mu^i_{P'}(a)\right)^2}{{\sigma^i_{P'}}^2}\right) \cdot \exp\left(-\frac{\left(X^i_{t+1}(a) - o^i_{t+1}\right)^2}{{\sigma^i_{ROM}}^2}\right) \\
&= \exp\left(\frac{-X^i_{t+1}(a)^2 + 2\mu^i_{P'}(a)X^i_{t+1}(a) - \mu^i_{P'}(a)^2}{{\sigma^i_{P'}}^2} + \frac{-X^i_{t+1}(a)^2 + 2o^i_{t+1}X^i_{t+1}(a) - {o^i_{t+1}}^2}{{\sigma^i_{ROM}}^2}\right) \\
&= c' \cdot \exp\left(-\frac{(X^i_{t+1}(a) - \mu^i_{P_{new}})^2}{{\sigma^i_{P_{new}}}^2}\right) = P^i_{new}(X^i_{t+1}(a)).
\end{aligned} \tag{3.14}
$$

In the case of multiple observations $O^i_{t+1} = o^{i,1}_{t+1}, o^{i,2}_{t+1}, \ldots, o^{i,|O^i_t|}_{t+1}$ we may also do the computation of the posterior for a single random variable $X^i_{t+1}(a)$ and use the resulting posterior for all other groundings of the relational atom (to which the same set of observations applies). The calculation is similar to the above, except that multiple observations need to be considered,

$$
\begin{aligned}
&P'^i\left(X^i_{t+1}(a)\right) \cdot \prod_{o \in O^i_{t+1}} \phi_{ROM}\left(o|X^i_{t+1}(a)\right) \\
&= \exp\left(-\frac{(X^i_{t+1}(a) - \mu^i_{P'}(a))^2}{{\sigma^i_{P'}}^2}\right) \cdot \exp\left(-\sum_{o \in O^i_{t+1}} \frac{(X^i_{t+1}(a) - o)^2}{{\sigma^i_{ROM}}^2}\right) \\
&= c'' \cdot \exp\left(-\frac{(X^i_{t+1}(a) - \mu^i_{P_{new}}(a))^2}{{\sigma^i_{new}}^2}\right) = P^i_{new}\left(X^i_{t+1}(a)\right).
\end{aligned} \tag{3.15}
$$

3.3.3 Lifted Inference with Individual Observations

One of the key challenges in lifted inference is handling individual observations. Current methods ground a relational atom when different observations are made for its random variables. It is usually the case that models shatter combinatorially fast and thus forfeit the benefits of a relational representation and the applicability of lifted inference.

We solve this problem in the LRKF by noting that the variances and covariances in the model are not affected by individual observations. We are thus able to represent the variances and covariances in a relational way while allowing variables to carry individual means. Further, the lifted prediction operation applies unmodified to this representation.

Lemma 7 *The variances of two random variables $X(a)$, $X(b)$ in an RGM are equal after a filtering step (Lifted Prediction and Lifted Update) if the following conditions hold before the filtering step: (1) both random variables are in the same relational atom; (2) the variance of both variables is the same; (3) observations are made for both variables or none of them.*

Proof Given conditions (1) and (2), we first prove that the variance of both random variables is the same after the Lifted Prediction step. Note that condition (3) is not relevant to this step.

WLOG we assume $X_t(a)$ and $X_t(b)$ have different means, $\mu_t(a)$ and $\mu_t(b)$. Moreover, it is easy to see that the variance of $X^i_{t+1}(a)$ and $X^i_{t+1}(b)$ is the same after marginalizing all random variables of timestep t due to the following two reasons: (i) $X(a)$ and $X(b)$ are in the same relational atom and thus share the same relationships with other random variables; (ii) the means are not involved in the marginalizations (see Section 3.3.1). It follows that we can represent the potentials

relevant to the marginalization of $X_t(a)$ and $X_t(b)$ as follows:

$$
\begin{aligned}
&\exp\left(-\frac{(X_t(a)-\mu_t(a))^2}{\sigma^2_{X_t(a)}}\right)\phi_{RTM}\left(X_{t+1}(a)|X_t(a),U_t(a)\right)\phi_{RPM}\left(X_t(a),X_t(b)\right)\\
&\exp\left(-\frac{(X(b)_t-\mu_t(b))^2}{\sigma^2_{X_t(b)}}\right)\phi_{RTM}\left(X_{t+1}(b)|X_t(b),U_t(b)\right)\phi_{RPM}\left(X(a)_{t+1},X(b)_{t+1}\right)\\
&=\exp\left(\mathbf{c}_{X_t(a)^2}X_t(a)^2+\mathbf{c}_{X_t(a)}X_t(a)\right)\exp\left(\frac{2B^i_X}{\sigma^2_{RTM}}X_t(a)X_{t+1}(a)\right)\exp\left(\frac{X_t(a)X_t(b)}{\sigma^2_{RPM}}\right)\\
&\quad\exp\left(\mathbf{c}_{X_t(b)^2}X_t(b)^2+\mathbf{c}_{X_t(b)}X_t(b)\right)\exp\left(\frac{2B^i_X}{\sigma^2_{RTM}}X_t(b)X_{t+1}(b)\right)\\
&\quad\cdot\phi_{\text{other}}\left(X_{t+1}(a),X_{t+1}(b)\right),
\end{aligned}
$$

where $\mathbf{c}_X$ refers to the coefficient of the term X.[4]

After $X_t(a)$ and $X_t(b)$ are marginalized we get a potential on $X_{t+1}(a)$ and $X_{t+1}(b)$. The variances of the random variables are the inverses of the coefficients of their squares in the resulting potential. Thus, all we need to show is that the coefficients of the square of the random variables, $X_{t+1}(a)^2$ and $X_{t+1}(b)^2$, are the same after marginalization. The two coefficients can be represented as follows,

$$
\mathbf{c}_{X_{t+1}(a)^2}=\frac{-\mathbf{c}_{X_t(b)^2}\left(\frac{B^i_X}{\sigma^2_{RTM}}\right)^2}{\left(\frac{1}{\sigma^2_{RTM}}\right)^2-\mathbf{c}_{X_t(a)}\,\mathbf{c}_{X_t(b)}},\ \mathbf{c}_{X_{t+1}(b)^2}=\frac{-\mathbf{c}_{X_t(a)^2}\left(\frac{B^i_X}{\sigma^2_{RTM}}\right)^2}{\left(\frac{1}{\sigma^2_{RTM}}\right)^2-\mathbf{c}_{X(a)_t}\,\mathbf{c}_{X_t(b)}}
$$

where, $\mathbf{c}_{X_t(\cdot)^2}=-\left(\frac{1}{\sigma^2_{X_t(\cdot)}}+\frac{1}{\sigma^2_{RTM}}+\frac{1}{\sigma^2_{RPM_t}}\right)$.

Condition (2) ($\sigma^2_{X_t(a)}$=$\sigma^2_{X_t(b)}$) implies $\mathbf{c}_{X_t(a)^2}=\mathbf{c}_{X_t(b)^2}$ which in turn implies $\mathbf{c}_{X_{t+1}(a)^2}$ = $\mathbf{c}_{X_{t+1}(b)^2}$. This is enough to prove that the variance of two random variables $X(a)$ and $X(b)$ with different means is the same after the Lifted Prediction step.

We now prove the result for the Lifted Update step. Regarding condition (3) there are two cases: (a) observations were made for both variables; or (b) no observations were made for either variable. In the case of (b) the proof is complete.

[4]For the sake of exposition the RTMs here represent dependences from state variables at time t to the same state variable at time $t+1$ (e.g. from $X_t(a)$ to $X_{t+1}(a)$). However, the general RTMs (e.g dependences from $X_t(a)$ to $X_{t+1}(b)$) produce similar forms.

In the case of (a), the update step for $X(a)$ can be represented by,

$$\exp\left(-\frac{(X_{t+1}(a)-\mu_{X_{t+1}}(a))^2}{\sigma^2_{X_{t+1}(a)^i}}-\frac{(X_{t+1}(a)-o_{a_t})^2}{{\sigma_{X_{ROM}}}^2}\right)=\exp\left(-\frac{(X_{t+1}(a)-{\mu^+}_{X_{t+1}}(a))^2}{{\sigma^+}_{X_{t+1}}(a)^2}\right)$$

where,

$${\sigma^{+2}}_{X_{t+1}(a)}=\frac{\sigma^2_{X_{t+1}(a)}\sigma^2_{X_{ROM}}}{\sigma^2_{X_{t+1}(a)^i}+\sigma^2_{X_{ROM}}},\ {\mu^+}_{X_{t+1}(a)}=\frac{\sigma^2_{X_{ROM}}\mu_{X_{t+1}(a)}+\sigma^2_{X_{t+1}(a)}o_{a_t}}{\sigma^2_{X_{ROM}}+\sigma^2_{X_{t+1}(a)}}$$

Likewise, after the update step the variance of $X(b)$ is,

$${\sigma^{+2}}_{X_{t+1}(b)}=\frac{\sigma^2_{X_{t+1}(b)^i}\sigma^2_{X_{ROM}}}{\sigma^2_{X_{t+1}(b)^i}+\sigma^2_{X_{ROM}}}$$

By condition (2) and the proof for the prediction step, $\sigma_{X_{t+1}(a)^i}=\sigma_{X_{t+1}(b)^i}$. Thus, ${\sigma^+}_{X_{t+1}(a)}={\sigma^+}_{X_{t+1}(b)}$. ■

Lemma 8 *The covariances of two pairs of variables ($X(a)$, $X(b)$) and ($X(a)$, $X(c)$) in an RGM are equal after a filtering step (Lifted Prediction and Lifted Update) if the following conditions hold before the filtering step: (1) the three random variables are in the same relational atom; (2) the covariance of both pairs of variables is the same; (3) observations are made for the three variables or none of them.*

Proof The method used in the proof of Lemma 7 can be employed in this proof: The terms involving the individual observations do not affect terms which determine the covariance of two random variables. ■

3.4 Algorithms and Computational Complexity

Let $\mathbb{X}$ ($|\mathbb{X}|$) be the set (number) of all random variables in the model and $X = (X^1, \ldots, X^{|X|})$ be the set of relational atoms (also, a partition of $\mathbb{X}$). In this section we speak of the relational atoms as sets of random variables.

Figure 3.3 presents our Lifted Relational Kalman filtering algorithm. The inputs to the algorithm are: relational atoms, X; the RGM, RTMs M_X, RPMs M_P and ROMs M_O; the prior over the relational atoms, P_0; and the control-inputs, $U_{[1,\ldots,T]}$, and observations, $O_{[1,\ldots,T]}$, for each timestep.

The algorithm computes the posterior recursively. **Split** partitions the domains of each relational atom X^i as induced by the control-inputs U_t. **Lifted_Predict** calculates new RPMs, M_P[5], and intermediate posterior, P_{int}, based on the transition models, M_X, and the control-inputs, U_t. Then, **Split_Obs** partitions the domains of each relational atom X^i as induced by the observations, O_t^i. **Lifted_Update** calculates the new posterior, P_{cur}, based on the intermediate posterior, P_{int}, the observation models, M_O, and the observations, O_t^i.

Given the control-inputs, **Split** partitions relational atoms as done in previous work: e.g. *Split* [Poole, 2003] and *SHATTER* [de Salvo Braz *et al.*, 2005]. If the control-inputs are allowed to differ for the variables in a relational atom, the model will be propositionalized. Hence, there is little advance in how we handle individual control-inputs with respect to previous algorithms [Choi *et al.*, 2010a].[6]

Algorithm **Split_Obs** partitions a relational atom X^i based on the observations. However, **Split_Obs** will only partition a relational atom in case the conditions of Lemmas 7 and 8 do not hold, i.e., when different number of observations are made for the relational variables. If the conditions of Lemmas 7 and 8 hold, the efficiency of the relational representation will be preserved even if multiple observations are made for all variables in some or all of the relational atoms.

Lemma 9 *The complexity of* **Lifted_Predict** *is* $O(|\mathbb{X}| \cdot |X_{t^+}|^2)$. *Where* X_{t^+} *is the set of relational atoms output by* **Split**.

Proof This step corresponds to the marginalization (Equation (3.13) and Ap-

[5] In our representation the number of relational atoms determines the number of RPMs which is equal to $E(|X|, 2)$ (the number of 2-combinations of $|X|$ with repetition).

[6] However, we conjecture that techniques similar to the ones we used for ROMs can be applied to RTMs. Any two random variables in the same atom will have the same variance after the Lifted_Predict step if they receive the same types of control inputs. That is, RTMs of the same type will increase the variances of the random variables by the same amount.

PROCEDURE **LRKF**(X,M_X,M_P,M_O, P_0, $U_{[1,\ldots,T]}$, $O_{[1,\ldots,T]}$)
Atoms, $X = (X^1, \ldots, X^{|X|})$; RTM, M_X, RPM, M_P, and ROM M_O;
prior, P_0; control-inputs, $U_{[1,\ldots,T]}$; observations, $O_{[1,\ldots,T]}$.

1. $P_{cur} \leftarrow P_0$, $X_{cur} \leftarrow X$
2. **For** $t = 1$ to T
   (a) $[X_{cur}, M_X, M_P, M_O] \leftarrow$ **Split**(X_{cur}, U_t, M_X, M_P, M_O)
   (b) $[P_{int}, M_p] \leftarrow$ **Lifted_Predict**(X_{cur}, P_{cur}, M_X, M_P, U_t) (§3.3.1)
   (c) $[X_{cur}, M_O] \leftarrow$ **Split_Obs**(X_{cur}, O_t, M_O) (§3.3.3)
   (d) $[P_{cur}] \leftarrow$ **Lifted_Update**(X_{cur}, M_O, P_{int}, O_t) (§3.3.2)
3. **Return** X_{cur}, P_{cur}

Figure 3.3: Algorithm Lifted_Relational_Kalman_Filter for Relational Gaussian Models.

pendix 3.8) of the variables in $\mathbb{X}$. For every variable that is integrated the parameters of all, $E(|X_{t^+}|, 2)$, pairwise interactions between relational atoms must be updated. ■

Lemma 10 *The complexity of* **Lifted_Update** *is* $O(|X_{t^{+o}}| \cdot |O_{max}|)$. *Where* $X_{t^{+o}}$ *is the set of relational atoms output by* **Split_Obs** *and* O_{max} *is the largest set of observations associated with a relational atom.*

Proof For each relational atom in $X_{t^{+o}}$ the computation in Equation (3.15) iterates over all relevant observations. ■

Our main result follows,

Theorem 11 *The computational complexity of* **LRKF** *is* $O(T \cdot (|\mathbb{X}| \cdot |X^*_{t^+}|^2 + |X^*_{t^{+o}}| \cdot |O^*_{max}|))$ *where* T *is the number of timesteps,* $\mathbb{X}$, $X^*_{t^+}$, $X^*_{t^{+o}}$ *and* O^*_{max} *are as above with the* $*$ *indicating the largest set across all timesteps.*

3.5 Related Work

The KF [Kalman, 1960; Roweis and Ghahramani, 1999] is a method for estimating the state of a dynamic process given a sequence of noisy observations.

It is restricted to linear dynamic and linear measurement models both with additive Gaussian noise. The Extended Kalman filter (EKF) [Sorenson and Stubberud, 1968] extends the KF to non-linear systems. For high dimensional data, a sampling method has been devised, the Ensemble Kalman filter [Evensen, 1994]. Exact Kalman filtering for high dimensional data is not feasible because exact filtering requires matrix inversions which take time cubic in the number of random variables.

Our RGMs represent the probability density as a product of node and edge factors. Any multivariate Gaussian is a quadratic exponential and can thus be written in this form. This is related to the information form of the Gaussian density and is the basis of other models such as Directed Gaussian Models (DGMs) [Cowell, 1998] and Gaussian Markov Random Fields (GMRFs) [Rue and Held, 2005]. However, RGMs are relational while DGMs and GMRFs are not. Thus, the previous models do not have a compact (relational) representations and, more importantly, an efficient (lifted) exact inference algorithm.

Relational probabilistic models allow the specification of models with size independent of the sizes of the populations in the model [Friedman *et al.*, 1999; Poole, 2003; Milch *et al.*, 2005; Richardson and Domingos, 2006]. Lifted inference algorithms [de Salvo Braz *et al.*, 2005; Milch and Russell, 2006] attempt to carry as much of the computations without propositionalizing the model. [Poole, 2003], solves inference problems by dynamically splitting and unifying sets of ground atoms. [de Salvo Braz *et al.*, 2005] (FOVE) introduced *counting elimination* to efficiently eliminate atoms with different parameterizations. [Milch *et al.*, 2008] (C-FOVE) take a slightly different approach with the introduction of counting formulas. However, all of the above lifted inference algorithms are not applicable to models with continuous variables.

[Kersting *et al.*, 2006] introduced Logical HMMs that combine ideas from *Statistical Relational Learning* and dynamic models. Indeed their work, as ours, pursues the benefits that the relational approach brings to inference and learning. However, their work is inherently discrete and further, they assume specific tran-

sition and observation models.

For relational models with continuous variables, recent advances have made inference possible. [Wang and Domingos, 2008] is an approximate algorithm based on sampling, search and local optimization. [Choi *et al.*, 2010a] is an exact variable elimination algorithm for continuous domains. The latter algorithm is similar to the marginalization problem that is part of the prediction step in filtering. However, none of these algorithms have been devised with dynamic models in mind nor do they address the problem of individual observations.

3.6 Experimental Results

We compare the average filtering time of **LRKF** and a conventional Kalman filter by varying the number of random variables. We implemented both the LRKF and the conventional KF (which handles random variables individually) in Perl. This makes the manipulation of the dynamically changing structure convenient.

For the housing market model in Figure 3.1, we randomly choose the parameters of the models (priors, RTMs, RPMs, and ROMs) and provide observations for HMO_t and $HPO_t(\cdot)$. To emphasize the difference in scalability, we assume that some set of houses has individual observations in each timestep, $HPO_t(\cdot)$, while the rest of the houses do not. We ran the two filters over 50 timesteps. The results in graph 3.4 confirm our theoretical results contrasting the linear time complexity of **LRKF** with the cubic time complexity of the Kalman filter.

3.7 Conclusion

We propose Relational Gaussian Models to represent and model dynamic systems in a relational (first-order) way. Further, we present the first algorithm for filtering or tracking at the first-order level. Our theoretical analysis and empirical tests show that our approach leads to significant gains in efficiency and enables filtering

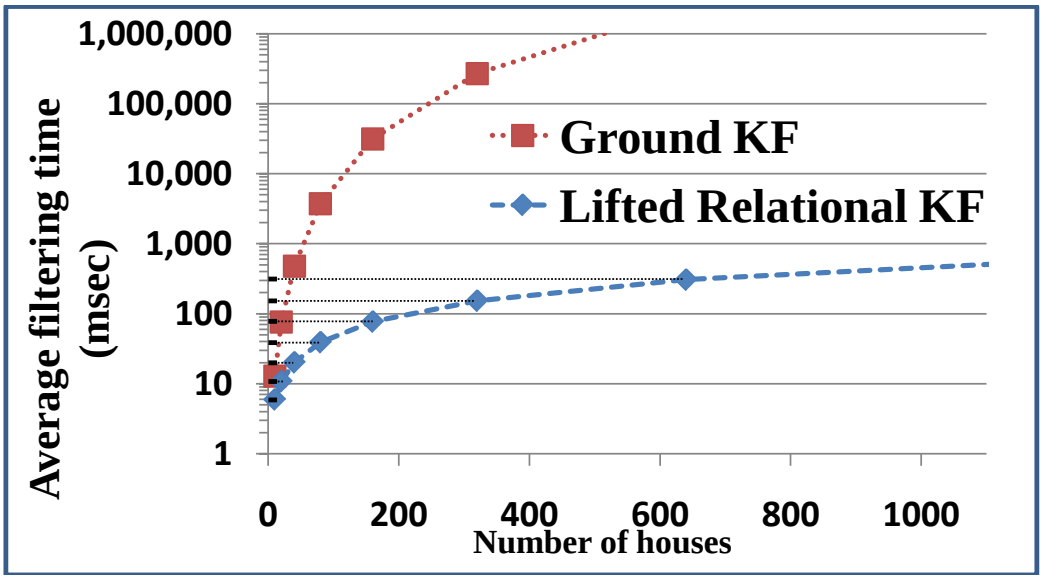


Figure 3.4: Average filtering time with increasing number of houses. Note the cubic increase in filtering time for the Ground Kalman filter and the linear increase for our Lifted Relational Kalman filter (LRKF). The y-axis is shown in logarithmic scale. To show that LRKF performs linearly, we added markers at the measurements on the LRKF curve.

for systems with very large numbers of random variables. We also make the case for the applicability of lifted inference to address real-world problems by taking a recently proposed model of social relationship strength and extending it to large dynamic networks.

3.8 Appendix: Details of Lifted Prediction

The integration is done using the following rule,

$$\int_{X_t^i(a^*)} \exp\left(-\mathbb{A}X_t^i(a^*)^2 + 2\mathbb{B}X_t^i(a^*) - \mathbb{C}\right) = \frac{\sqrt{\pi}}{\sqrt{\mathbb{A}}} \exp\left(\frac{\mathbb{B}^2}{\mathbb{A}} - \mathbb{C}\right). \tag{3.16}$$

where $\mathbb{A}$ is a constant, $\mathbb{B}$ a linear form of random variables except $X_t^i(a^*)$, and $\mathbb{C}$ is a quadratic form of random variables except $X_t^i(a^*)$.

The integration of one random variable in Equation (3.12) can be represented

as follows,

$$\int_{X_t^i(a^*)} \prod_{1\le j\le n} \prod_{a'\in A^j} \exp\left(-\frac{\left(X_{t+1}^j(a') - B_X^{i,j} X_t^i(a^*) - B_U^{i,j} U_t^i(a^*)\right)^2}{{\sigma_{RTM}^i}^2}\right) \exp\left(-\frac{\left(X_t^i(a^*) - \mu_p^i(a^*)\right)^2}{{\sigma_P^i}^2}\right)$$

$$\cdot \exp\left(-\frac{\left(X_t^i(a^*) - R_t^{i,j} X_t^j(a')\right)^2}{{\sigma_{RPM}^{i,j}}^2}\right) \tag{3.17}$$

$$= \int_{X_t^i(a^*)} \exp\left(-\mathbb{A} X_t^i(a^*)^2 + \left(\mathbf{c} + \sum_{1\le j\le n} \sum_{a'\in A^j} \mathbf{c}_t^j X_t^j(a') + \mathbf{c}_{t+1}^j X_{t+1}^j(a')\right) X_t^i(a^*) - \mathbb{C}\right)$$

$$= \int_{X_t^i(a^*)} \exp\left(-\mathbb{A} X_t^i(a^*)^2 + \left(\mathbf{c} + \sum_{1\le j\le n} \mathbf{c}_t^j \sum_{a'\in A^j} X_t^j(a') + \mathbf{c}_{t+1}^j \sum_{a'\in A^j} X_{t+1}^j(a')\right) X_t^i(a^*) - \mathbb{C}\right)$$

$$= \int_{X_t^i(a^*)} \exp\left(-\mathbb{A} X_t^i(a)^2 + \left(\mathbf{c} + \sum_{1\le j\le n} \mathbf{c}_t^j \mathbb{X}_t^j + \mathbf{c}_{t+1}^j \mathbb{X}_{t+1}^j\right) X_t^i(^*a) - \mathbb{C}\right), \tag{3.18}$$

when $\mathbf{c}$, $\mathbf{c}_t^j$ and $\mathbf{c}_{t+1}^j$ represent constants calculated from Equation (3.17), and $\mathbb{X}_t^j$ represents $\sum_{a'\in A^j} X_t^j(a')$.

Note the quadratic form in Equation (3.16) includes the following types of expression,

$$(\mathbb{X} + \mathbb{X}')^2 = [\mathbb{X}^2] + 2[\mathbb{X}\mathbb{X}] + 2\mathbb{X}\mathbb{X}' + [\mathbb{X}'^2] + 2[\mathbb{X}'\mathbb{X}'], \tag{3.19}$$

where $[\mathbb{X}^2]$ is $\sum_{a\in A} X(a)^2$, and $[\mathbb{X}\mathbb{X}]$ is $\sum_{a,a'\in A, a\neq a'} X(a)X(a')$.

Now, Equation (3.18) is integrated as follows,

$$
\begin{aligned}
&\frac{\sqrt{\pi}}{\sqrt{\mathbb{A}}}\exp\left(\frac{1}{\mathbb{A}}\left(\mathbf{c}+\sum_{1\le j\le n}\mathbf{c}_t^j\mathbb{X}_t^j+\mathbf{c}_{t+1}^j\mathbb{X}_{t+1}^j\right)^2-\mathbb{C}\right)\\
&=\frac{\sqrt{\pi}}{\sqrt{\mathbb{A}}}\exp\left(\frac{1}{\mathbb{A}}\sum_{1\le j\le n}\left(\mathbf{c}_t^{j^2}\left[\mathbb{X}_t^{j^2}\right]+2\mathbf{c}_t^{j^2}\left[\mathbb{X}_t^j\mathbb{X}_t^j\right]+2\mathbf{c}\mathbf{c}_t^j\mathbb{X}_t^j\right)\right)\cdot\exp\left(-\mathbb{C}\right)\\
&\cdot\exp\left(\frac{1}{\mathbb{A}}\sum_{1\le j\le n}\left(\mathbf{c}_{t+1}^{j\ 2}\left[\mathbb{X}_{t+1}^{j\ 2}\right]+2\mathbf{c}_{t+1}^{j\ 2}\left[\mathbb{X}_{t+1}^j\mathbb{X}_{t+1}^j\right]+2\mathbf{c}\mathbf{c}_{t+1}^j\mathbb{X}_{t+1}^j\right)\right)\\
&\cdot\exp\left(\frac{1}{\mathbb{A}}\sum_{1\le j<j'\le n}\left(2\mathbf{c}_t^j\mathbf{c}_t^{j'}\mathbb{X}_t^j\mathbb{X}_t^{j'}+2\mathbf{c}_t^j\mathbf{c}_{t+1}^{j'}\mathbb{X}_t^j\mathbb{X}_{t+1}^{j'}+2\mathbf{c}_{t+1}^j\mathbf{c}_{t+1}^{j'}\mathbb{X}_{t+1}^j\mathbb{X}_{t+1}^{j'}\right)\right) \qquad (3.20)\\
&=\prod_{1\le j<j'\le n}\ \prod_{\substack{a\in A^j,a'\in A^{j'}\\ a\neq a^* \text{ if } i=j\\ a'\neq a^* \text{ if } i=j'}}\exp\left(-\frac{\left(X_t^i(a)-R_t'^{j,j'}X_t^{j'}(a')\right)^2}{\sigma'^{\ j,j'\ 2}_{tRPM}}\right)\exp\left(-\frac{\left(X_t^j(a)-\mu_{t\,P'}^{\ j}(a)\right)^2}{\sigma_{t\,P'}^{\ j\ 2}}\right)\\
&\cdot\prod_{1\le j<j'\le n}\ \prod_{a\in A^j,a'\in A^{j'}}\exp\left(-\frac{\left(X_{t+1}^j(a)-R_{t+1}'^{j,j'}X_{t+1}^{j'}(a')\right)^2}{\sigma'^{\ j,j'\ 2}_{t+1RPM}}\right)\exp\left(-\frac{\left(X_{t+1}^j(a)-\mu_{t+1\,P'}^{\ j}(a)\right)^2}{\sigma_{t+1\,P'}^{\ j\ 2}}\right)\\
&\cdot\prod_{1\le j,j'\le n}\ \prod_{\substack{a\in A^j,a'\in A^{j'}\\ a'\neq a^* \text{ if } i=j'}}\exp\left(-\frac{\left(X_{t+1}^j(a)-R_{t,t+1}'^{j,j'}X_t^{j'}(a')\right)^2}{\sigma'^{\ j,j'\ 2}_{t,t+1RPM}}\right).
\end{aligned}
$$

Here, R'_t, R'_{t+1}, $R'_{t,t+1}$, μ_t, μ_{t+1}, σ'_{tRPM} and σ'_{t+1RPM} are new constants derived from Equation (3.20).

CHAPTER 4

LIFTED INFERENCE WITH AGGREGATE FACTORS

Aggregate factors (that is, those based on aggregate functions such as *SUM*, *AVERAGE*, *AND* etc) in probabilistic relational models can compactly represent dependencies among a large number of relational random variables. However, propositional inference on a factor aggregating n k-valued random variables into an r-valued result random variable is $O(rk2^n)$. Lifted methods can ameliorate this to $O(rn^k)$ in general and $O(rk \log n)$ for commutative associative aggregators. In this chapter, I propose (a) an *exact* solution *constant* in n when k=2 for certain aggregate operations such as *AND*, *OR* and *SUM*, and (b) a close approximation for inference with aggregate factors with time complexity *constant* in n. This approximate inference involves an analytical solution for some operations when k>2. The approximation is based on the fact that the typically used aggregate functions can be represented by linear constraints in the standard $(k-1)$-simplex in $\mathbb{R}^k$ where k is the number of possible values for random variables. This includes even aggregate functions that are commutative but not associative (e.g., the *MODE* operator that chooses the most frequent value). Our algorithm takes polynomial time in k (which is only 2 for binary variables) regardless of r and n, and the error decreases as n increases. Therefore, for most applications (in which a close approximation suffices) our algorithm is a much more efficient solution than existing algorithms. I present experimental results supporting these claims. I also present a (c) third contribution which further optimizes aggregations over multiple groups of random variables with distinct distributions.

4.1 Introduction

Relational models can compactly (that is, intensionally) represent graphical models involving a large number of random variables, each of them representing a relation between objects in a domain [Koller and Pfeffer, 1997; Friedman *et al.*, 1999; Milch *et al.*, 2005; Richardson and Domingos, 2006].

While it is possible to take advantage of compactness only for representation and expand the model into a propositional (extensional) form for inference, lifted inference methods try to keep the representation as compact as possible even during inference, increasing efficiency [Poole, 2003; de Salvo Braz *et al.*, 2007; Milch *et al.*, 2008; Singla and Domingos, 2008].

The first proposed lifted inference solutions could deal only with factors on a fixed number of random variables. *Aggregate* parametric factors (based on aggregate functions such as *OR*, *MAX*, *AND*, *SUM*, *AVERAGE*, *MODE* and *MEDIAN*), which are defined on a varying, intensionally defined set of random variables, still needed to be treated propositionally, with cost exponential in the number n of random variables. [Kisynski and Poole, 2009] introduced lifted methods for aggregate factors that reduce this complexity to $O(rk \log n)$ for commutative associative aggregate functions on n k-valued random variables being aggregated into an r-valued random variable (and even $O(rk)$ for *OR* and *MAX*)[1]. However, for general cases (such as the non-associative function *MODE*), their exact inference method has time $O(rn^k)$, that is, polynomial in n.

The contributions of this chapter are threefold. I contribute an *exact* solution *constant* in n when $k = 2$ for aggregate operations *AND*, *OR*, *MAX* and *SUM*. I also present an efficient (*constant* in n) approximate algorithm for inference with aggregate factors, for all typical aggregate functions. The potential of a aggregate factor for a valuation v of a set of random variables depends only on the *histogram* on the distribution of k values in V (in what [Milch *et al.*, 2008] calls a *counting*

[1] Note that r=n for aggregate functions such as *SUM* of n binaray variables.

formula). I show that the typical aggregate functions but for *XOR*[2] can be represented by linear constraints in the space of histograms (a $(k-1)$-simplex). Because aggregate factors' potentials on the space of histograms can be approximated by a normal distribution, one can approximately sums over them (which is the main inference operation) by computing the volume under normal distributions truncated by linear constraints. This holds even for *MODE*, which is commutative but not associative.

This approximation can be computed analytically for all operations on binary random variables and for certain operations on multivalued (k>2) random variables such as *SUM* and *MEDIAN*. Otherwise, it is computed by Gibbs sampling with a limited number of iterations [Geweke, 1991; Damien and Walker, 2001]. Finally, a third contribution is a further optimization for aggregations of multiple groups of random variables, each with its own distribution.

This chapter is organized as follows. Section 4.2 defines relational models and our inference problem, *AFM* (Aggregation Factor Marginalization). Section 4.3 presents our lifted inference methods for aggregate factors followed by an extended algorithm for the generalized problems in Section 4.4. Section 4.5 provides the error bounds of the approximations. I present some empirical results in Section 4.6. The chapter concludes in Section 4.7.

4.2 Background and Problem Definition

This chapter is concerned about inference problems over relational models with aggregate factors. I now revisit these concepts.

4.2.1 First-order Probabilistic Models

A **factor** f is a pair (A_f, ϕ_f) where A_f is a tuple of random variables and ϕ_f is a **potential function** from the range of A_f to the nonnegative real numbers. Given a

[2] *XOR* has its own simple solution.

valuation v of random variables (**rv**s), the **potential** of f on v is $w_f(v) = \phi_f(A_f)$.

The joint probability defined by a set F of factors on a valuation v of random variables is the normalization of $\prod_{f \in F} w_f(v)$. If each factor in F is a conditional probability of a child random variable given the value of its parent random variables, and there are no directed cycles in the graph formed by directed edges from parents to children, then the model defines a Bayesian network. Otherwise it is an undirected model.

One can have parameterized (indexed) random variables by using **predicates**, which are functions mapping parameter values (indices) to random variables. A **relational atom** is an application of a predicate, possibly with free variables. For example, a predicate $friends$ is used in atoms $friends(X, Y)$, $friends(X, bob)$ and $friends(john, bob)$, where X and Y are free variables and $john$ and bob possible parameter values. $friends(john, bob)$ is a **ground atom** and directly corresponds to a random variable.

A **parfactor** is a tuple (L, C, A, ϕ) composed of a set of parameters (also called *logical variables*) L, a constraint C on L, a tuple of atoms A, and a potential function ϕ. Let a **substitution** θ be an assignment to L and $A\theta$ the relational atom (possibly ground) resulting from replacing logical variables by their values in θ. A parfactor g stands for the set of factors $gr(g)$ with elements $(A\theta, \phi)$ for every assignment θ to the parameters L that satisfies the constraint C. A First-order Probabilistic Model (**FOPM**) is a compact, or intensional, representation of a graphical model. It is composed by a **domain**, which is the set of possible parameter values (referred to as **domain objects**) and a set of parfactors. The corresponding graphical model is the one defined by all instantiated factors. The joint probability of a valuation v according to a set of parfactors G is

$$P(v) = 1/Z \prod_{g \in G} \prod_{f \in gr(g)} w_f(v), \tag{4.1}$$

where Z is a normalization constant.

Example: The dependence between political ads and votes in the example in

Figure 1 can be compactly represented by the parfactor $(\{i\}, \top, (V(i), Ads), P(V(i)|Ads))$ with a domain formed by the set of voters ($\top$ represents a tautology, so no constraints are posed on i and instances are generated for all voters). The figure uses the more traditional notation V_i, equivalent to $V(i)$.

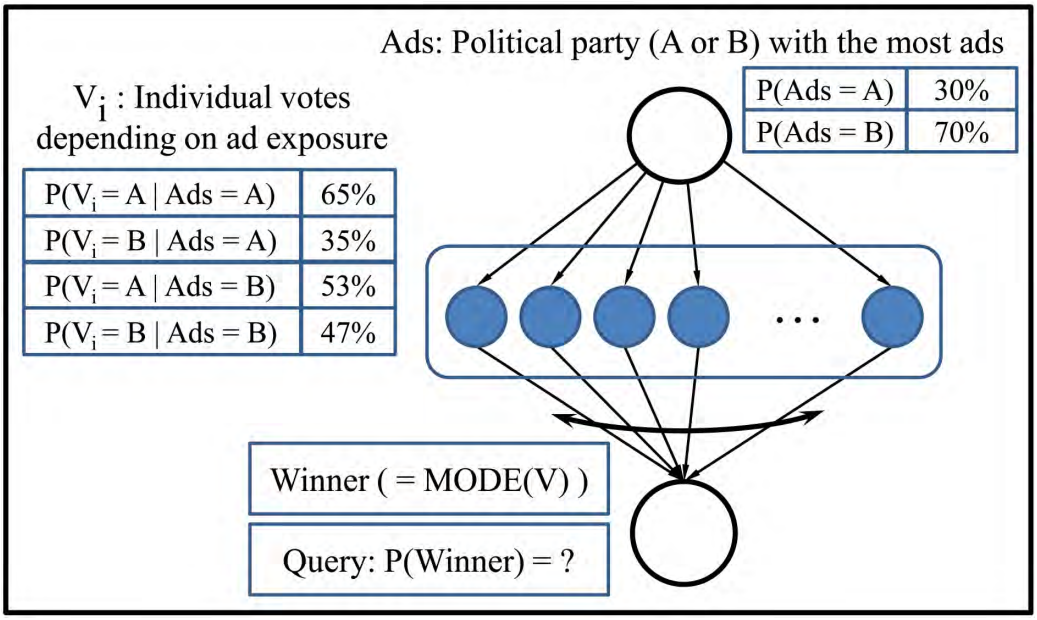


Figure 4.1: Graphical model on the domain of the election of one of two parties A and B. The random variable *Ads* indicates which party has the most ads in the media. The variables V_i indicate the vote of each person in a population, modeled as a dependence of ad exposure. The *Winner* variable indicates the winner and it is determined by the majority (*MODE*) of votes. One would like to estimate the probability of each party winning the election given this model.

.

4.2.2 Aggregate Factors and Parfactors

An **aggregate factor** is a factor $((X_1, \ldots, X_n, Y, \phi_\otimes))$ where $\phi_\otimes$ establishes that the valuation y of Y must be the result of an aggregation function $\otimes$ over the valuation $x_1, \ldots, x_n$ of $X_1, \ldots, X_n$:

$$\phi_\otimes(x_1, \ldots, x_n, y) = \begin{cases} 1 & \text{if } y = \bigotimes_{i=1,\ldots,n} x_i \\ 0 & \text{otherwise} \end{cases} . \tag{4.2}$$

This chapter considers the aggregate functions *OR*, *MAX*, *AND*, *XOR*, *SUM*, *AVERAGE*, *MODE* and *MEDIAN*. Noisy versions such as *Noisy-OR* can be represented by adding an extra factor on x_i.[3]

An **aggregate parfactor** $g = (L, C, X, \otimes, Y)$, where X and Y are now relational atoms, can be used by FOPMs to compactly represent a set of aggregate factors. The set $gr(g)$ of ground factors instantiated from g comprises the aggregate factors $((X\theta_0\theta_1, \ldots, X\theta_0\theta_n, Y\theta_0), \phi_\otimes)$, for each substitution θ_0 on the logical variables in Y consistent with constraint C, and substitutions $\theta_1, \ldots, \theta_n$ on the logical variables in X but not in Y consistent with C. For the example in Figure 1, the conditional probability of *Winner* can be compactly represented by the aggregate parfactor $(i, \top, V(i), MODE, Winner)$. More general aggregation cases (for example, with aggregated random variables sets including more than one predicate) can be normalized to this type of aggregated parfactor, as detailed in [Kisynski and Poole, 2009].

4.2.3 Inference with Aggregate Parfactors: Aggregate Factor Marginalization (AFM)

This section is concerned about the inference problem of marginalizing a set of rvs in an FOPM with aggregate factors to determine the marginal density of others. As shown by [Kisynski and Poole, 2009], this can be done by using C-FOVE [Milch *et al.*, 2008] extended with a lifted operation for summing random variables out of an aggregate parfactor. These summations can be reduced to the Aggregate Factor Marginalization (AFM) calculation:

$$\phi'_{\mathbf{y}}(y) = \sum_{x_1,,x_n} \left(\phi_\otimes(y, x_1, \ldots, x_n) \prod_{1 \le i \le n} \phi_{\mathbf{x}}(x_i) \right).$$

[3] Our definitions are based on [Kisynski and Poole, 2009] but differ from theirs in this aspect; while our aggregate factors are deterministic, theirs include an extra potential for noisy versions. As explained, one can do the same with an extra factor/parfactor.

where $\phi_{\mathbf{x}}$ is the (same for all i) potential product of all other factors in the model that have X_i as an argument, and $\phi'_{\mathbf{y}}$ is the resulting potential on y alone. This subproblem is also one that needs to be solved in extending Lifted Belief Propagation [Singla and Domingos, 2008] to deal with aggregate factors.

[Kisynski and Poole, 2009] shows how, when different x_i have different potential functions on them, the problem can be normalized (by splitting and using auxiliary variables) to multiple such sums in which this uniformity holds. Similarly, one can separate the case in which only *some* x_i need to be summed out into two different aggregate parfactors, one for all aggregate random variables being summed out, and another for the remaining ones.

A direct computation of **AFM** is exponential in n. [Kisynski and Poole, 2009] shows lifted operations that can be done in time polynomial or logarithmic in n (depending on certain conditions explained below). In Section 4.3 I present two lifted methods, one exact and one approximate, with time constant in n.

4.2.4 Inference Problems with Inequality

This section defines aggregate factors with inequality constraints by using

$$\phi_{\otimes_{\leq}}(y, x_1, \ldots, x_n) = \begin{cases} 1 & \text{if } y \leq x_1 \otimes \cdots \otimes x_n \\ 0 & \text{otherwise} \end{cases}$$

with the corresponding problem **AFM**[$\leq$] defined as

$$\sum_{x_1,\ldots,x_n} \left(\phi_{\otimes_{\leq}}(y, x_1, \ldots, x_n) \cdot \prod_{1 \leq i \leq n} \phi_{\mathbf{x}}(x_i) \right).$$

$\phi_{\otimes_{\geq}}$ and **AFM**[$\geq$] are defined analogously.

4.2.5 Existing Methods for AFM Problems

MAX and its special case *OR* (as well as their noisy versions) allow factorizations leading to lifted marginalization constant in n [Díez and Galán, 2003]. These operators can be decomposed into the product of n potentials:[4]

$$\sum_{x_1,\ldots,x_n} \phi_{\otimes}(y, x_1, \ldots, x_n) \cdot \prod_{i=1}^{n} \phi_{\mathbf{x}}(x_i) = \sum_{y'} \sum_{x_1,\ldots,x_n} \prod_{i=1}^{n} \phi_{\mathbf{y}',\mathbf{y}}(y', y) \cdot \phi_{\mathbf{y}',\mathbf{x}}(y', x_i) \tag{4.3}$$

$$= \sum_{y'} \left(\phi_{\mathbf{y}',\mathbf{y}}(y', y) \prod_{i=1}^{n} \sum_{x_i} \phi_{\mathbf{y}',\mathbf{x}}(y', x_i) \right). \tag{4.4}$$

Because the product is over a term independent of n, one can compute it once and exponentiate in time constant in n:

$$= \sum_{y'} \left(\phi_{\mathbf{y}',\mathbf{y}}(y', y) \left(\sum_{x'} \phi_{\mathbf{y}',\mathbf{x}}(y', x') \right)^{n} \right).$$

For other aggregate functions that happen to be commutative and associative, **AFM** can be computed by a recursive decomposition [Kisynski and Poole, 2009] into a subproblem with half the number of aggregated random variables, and therefore in time $O(r^2 k \log n)$ when n is a power of 2:

$$\sum_{x_1,\ldots,x_n} \phi_{\otimes}(y, x_1, \ldots, x_n) \prod_{i=1}^{n} \phi_{\mathbf{x}}(x_i) = \sum_{y = y' \otimes y''} \left(\sum_{x_1,\ldots,x_{\frac{n}{2}}} \phi_{\otimes}(y', x_1, \ldots, x_{\frac{n}{2}}) \prod_{i=1}^{\frac{n}{2}} \phi_{\mathbf{x}}(x_i) \right) \cdot \left(\sum_{x_{\frac{n}{2}+1},\ldots,x_n} \phi_{\otimes}(y'', x_{\frac{n}{2}+1}, \ldots, x_n) \prod_{i=\frac{n}{2}+1}^{n} \phi_{\mathbf{x}}(x_i) \right),$$

[4] See [Díez and Galán, 2003] for details on $\phi_{\mathbf{y}',\mathbf{y}}$ and $\phi_{\mathbf{y}',\mathbf{x}}$.

where $\phi_{\otimes}(y, x_i) = \begin{cases} 1 & \text{if } y = x_i \\ 0 & \text{otherwise} \end{cases}$.

Note that the two decomposition halves are the same problem up to variable renaming and thus computed in time $O(k \log n)$, r^2 times (once per value of y' or y'' and another per value of y). [Kisynski and Poole, 2009] describes the minor adjustments needed when n is not a power of 2.

4.3 Efficient Methods for AFM Problems

This section now presents solutions for **AFM** problems. The exact solutions presented in the previous section are efficient. However, their applicability is limited to some operations [Díez and Galán, 2003], or their computational complexity still depends on the number of rvs [Kisynski and Poole, 2009]. Here, it proposes an exact solution for some cases, and new efficient approximate marginalizations that are applicable to more aggregate functions.

4.3.1 Normal Distribution with Linear Constraints

[Kisynski and Poole, 2009] shows how the potential of an aggregate parfactor depends only on the value histogram of its aggregated random variables (histograms were introduced in Counting Elimination [de Salvo Braz *et al.*, 2007] and used as counting formulas in [Milch *et al.*, 2008]).

Given values $x_1, \ldots, x_n$ for n rvs with the same range, the value histogram of x is a vector h with $h_u = |\{i : x_i = u\}|$ for each u in the rvs' range. When a potential function on $x_1, \ldots, x_n$ depends on the histogram alone, as in the case of aggregate factors, then there is a function $\phi_{\mathbf{h}}$ on histograms such that $\phi(y, x_1, \ldots, x_n) = \phi_{\mathbf{h}}(y, h)$ and $\phi_{\otimes}(y, x_1, \ldots, x_n) = \phi_{\otimes \mathbf{h}}(y, h)$. In what follows, this section describes the binomial case (range of x_i equal to 2) for clarity, but it applies to the

multinomial case as well. One can write

$$\sum_{x_1,\ldots,x_n} \phi(y, x_1, \ldots, x_n) \prod_i \phi_{\mathbf{x}}(x_i) = \sum_h \binom{n}{h_1} \phi_{\mathbf{h}}(y, h) p_1^{h_1} p_0^{n-h_1}, \tag{4.5}$$

where p_0, p_1 are the normalizations of $\phi_{\mathbf{x}}$. This corresponds to grouping assignments on x into their corresponding histograms h, and iterating over the histograms (which are exponentially less many), taking into account that each histogram corresponds to $\binom{n}{h_1}$ assignments.

One now observes that functions $\phi_{\mathbf{h}}(y, h)$ coming from aggregate factors always evaluate to 0 or 1. Moreover, the set of histograms for which they evaluate to 1 can be described by linear constraints on the histogram components. For example, $\phi_{MODE}(y, h)$ will only be 1 if $h_y \geq h_{y'}$ for all $y' \neq y$. Given $\phi_{\mathbf{h}}$ and y, let C_y be the set of histograms h such that $\phi_{\mathbf{h}}(y, h)=1$. Then (4.5) can be rewritten as

$$\sum_{h \in C_y} \binom{n}{h_1} p_1^{h_1} p_0^{n-h_1},$$

which is the probability of a set of h_1 values under a binomial distribution. For large n, according to the Central Limit Theorem [Rice, 2006], the binomial distribution is approximated by the normal distribution $N(np_1, np_1p_0)$ with density function f. Then

$$\sum_{h \in C_y} \binom{n}{h_1} p_1^{h_1} p_0^{n-h_1} \approx \int_{h' \in C'_y} f(h')\, dh',$$

where C'_y is a continuous region in the $(k-1)$-simplex corresponding to C_y (which is defined in discrete space). Table 4.1 lists C_y and an appropriate C'_y for the several aggregate factor potentials, for both **AFM** and **AFM[≥]**.

Let's see two examples. For **AFM** on *MODE* on binary variables, $y = 1$, and histograms with $h(1) = t$, C_y is $h_1 \geq h_0$ and C'_y is $t \in \left[\left\lfloor \frac{n}{2} \right\rfloor + 0.5, n + 0.5\right]$[5], so one

[5]Here, +0.5 and −0.5 are continuity corrections for accurate approximations.

computes

$$\int_{t=\lfloor \frac{n}{2} \rfloor+0.5}^{n+0.5} f(t)\,dt,$$

which can be done in constant time. Let us also consider **AFM** and **AFM**[≥] on *SUM* with n=100 rvs representing ratings of 100 people who watch a movie. Each person gives ratings of either 0 (negative) or 1 (positive), with probabilities 0.55 and 0.45, respectively (p_0=0.55). One may be interested in the summation of those votes (r=100). Figure 4.2 shows the probability density of the number of positive ratings. The bars in red in (a) and (b) panels show the area corresponding to the result for **AFM** and **AFM**[≥], respectively, for y=50. The former can have the exact binomial distribution form computed in constant time, while the latter can have the normal distribution approximation computed in constant time. Therefore, the marginal on Y can be approximated in $O(r)$. [Kisynski and Poole, 2009]'s algorithm, on the other hand, takes $O(r \log n)$, and [Díez and Galán, 2003] is not applicable.

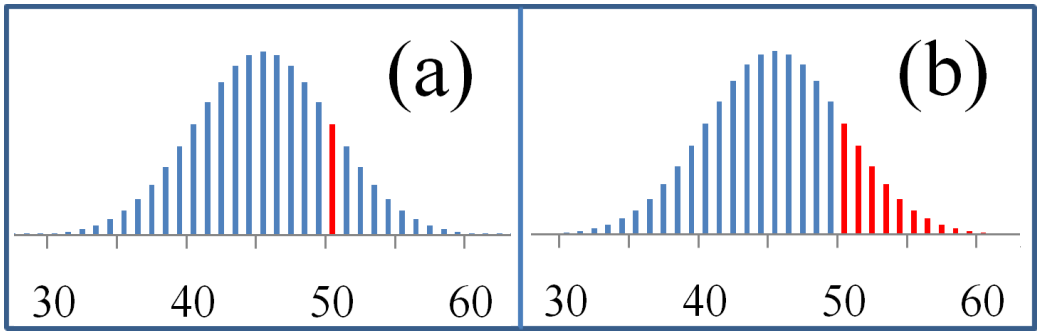


Figure 4.2: Histogram with a binomial distribution with (a) equality and (b) inequality constraints.

This chapter now explains the method in more detail for two different cases: aggregated binary random variables (k=2), which can be dealt with analytically, and aggregated multivalued random variables (k>2).

Operator	Problem	y	C_y	C'_y
AND	**AFM**	*TRUE*	$h_{TRUE} = n$	not needed (cheap exact solution)
OR	**AFM**	*FALSE*	$h_{FALSE} = n$	not needed (cheap exact solution)
SUM	**AFM**	y	$\sum_i i \times h_i = y$	$y - 0.5 \leq \sum_i i \times h_i \leq y + 0.5$
SUM	**AFM**[$\geq$]	y	$\sum_i i \times h_i \leq y$	$\sum_i i \times h_i \leq y - 0.5$
MAX	**AFM**	y	$h_y > 0$, $\forall i{>}y \;\; h_i = 0$	$h_y > 0.5$, $\forall i{>}y \;\; -0.5 \leq h_i \leq 0.5$
MAX	**AFM**[$\geq$]	y	$\forall i{>}y \;\; h_i = 0$	$\forall i{>}y \;\; -0.5 \leq h_i \leq 0.5$
MODE	**AFM**	y	$\forall i{\neq}y \;\; h_y > h_i$	$\forall i{\neq}y \;\; h_y > h_i$
MEDIAN	**AFM**	y	$\sum_{i=1}^{y-1} h(i) < \frac{n}{2}$, $\sum_{i=y}^{n} h(i) \geq \frac{n}{2}$	$\sum_{i=1}^{y-1} h(i) + 0.5 \leq \lfloor \frac{n}{2} \rfloor \leq \sum_{i=y}^{n} h(i) - 0.5$
MEDIAN	**AFM**[$\geq$]	y	$\sum_{i=1}^{y-1} h(i) \geq \frac{n}{2}$	$\sum_{i=1}^{y-1} h(i) - 0.5 \geq \lfloor \frac{n}{2} \rfloor$

Table 4.1: Constraints to be used in binomial (multinomial) distribution exact calculations (C_y) and (multivariate) Normal distribution approximations (C'_y). The table does not exhaust all combinations. However those omitted are easily obtained from the presented ones. E.g., $\phi_{OR}(T, x) = 1 - \phi_{OR}(F, x)$, $\phi_{AVERAGE}(y, x) = \phi_{SUM}(y \times n, x)$, and $\phi_{MODE\geq}(y, x) = \sum_{y' \leq y} \phi_{MODE}(y', x)$.

4.3.2 Binary Variables Case

AFM Problem

For *AND*, *OR*, *MIN*, *MAX* and *SUM*, an exact solution with time constant in n for **AFM** for the binary case can be computed, for the appropriate choices of p_0 and p_1, as

$$\phi'_{\mathrm{y}}(y) = \binom{n}{y} p_0^{n-y} \cdot {p_1}^y.$$

AVERAGE can be solved by using ϕ'_{y} obtained from *SUM* on y/n. This solution follows from the fact that, for the above cases, one needs the potential of a single histogram.

For *MODE* and *MEDIAN*, exact solutions for **AFM** are of the following form,

with time linear in n:

$$\phi'_{\mathbf{y}}(TRUE) = \sum_{i=\lfloor \frac{n}{2} \rfloor + 1}^{n} \binom{n}{i} p_0^{n-i} \cdot {p_1}^i.$$

Such solutions are more expensive because they measure the density of a region of histograms. They can be approximated by the Normal distribution in the following way:

$$\phi'_{\mathbf{y}}(TRUE) \approx \int_{t=\lfloor \frac{n}{2} \rfloor + 0.5}^{n+0.5} \frac{exp\left(-\frac{(t-np_1)^2}{2 \cdot np_1(1-p_1)}\right)}{\sqrt{2\pi \cdot np_1(1-p_1)}}\, dt.$$

Note that *MODE* is not solved by either [Díez and Galán, 2003]'s factorization or [Kisynski and Poole, 2009]'s logarithmic algorithm, while our approach can compute an approximation in constant time. For n is 100, $p_1 = 0.45$, the exact solution is about 0.18272. Our approximate solution is about 0.18286. Thus, the error is less than 0.1% of the exact solution.

AFM[$\leq$] and **AFM**[$\geq$] Problems

For binary aggregated random variables, these problems are different from **AFM** only for the *SUM* (and thus, *AVERAGE*) case. For *SUM* one can use the approximation

$$\phi'_{\mathbf{y}}(y) = \sum_{i=y}^{n} \binom{n}{i} p_1^i (1-p_1)^{n-i} \approx \int_{t=y-0.5}^{n+0.5} \frac{exp\left(-\frac{(t-np_1)^2}{2 \cdot np_1(1-p_1)}\right)}{\sqrt{2\pi \cdot np_1(1-p_1)}}\, dt.$$

4.3.3 Multivalued Variables Case

In the multivalued (k>2) case, there is a need to compute the probability of a linearly constrained region of histograms, which motivates us to consider approximate solutions with the multivariate Normal distribution. Consider the following

example: suppose that the aggregation function is *SUM*. There are 100 rvs representing ratings of 100 people who watch a movie. Each person gives ratings among 0, 1 and 2 (0 is lowest and 2 is highest). One may want to calculate the sum of ratings from 100 people when each person gives a rating 0 with 0.35 ($p(x_i{=}r0){=}0.35$), 1 with 0.35 ($p(x_i{=}r1){=}0.35$), and 2 with 0.3 ($p(x_i{=}r2){=}0.3$). The probability of histograms is provided by the multinomial distribution, as shown in Figure 4.3. The colored bars in (a) represent the probability of the ratings sum being exactly 100. If instead one wishes to determine the probability of the ratings sum exceeding 100, one may have an **AFM**[$\geq$] instance, with a probability corresponding to the colored bars in the (b) panel. In both cases, we need to compute the volume of a histogram region.

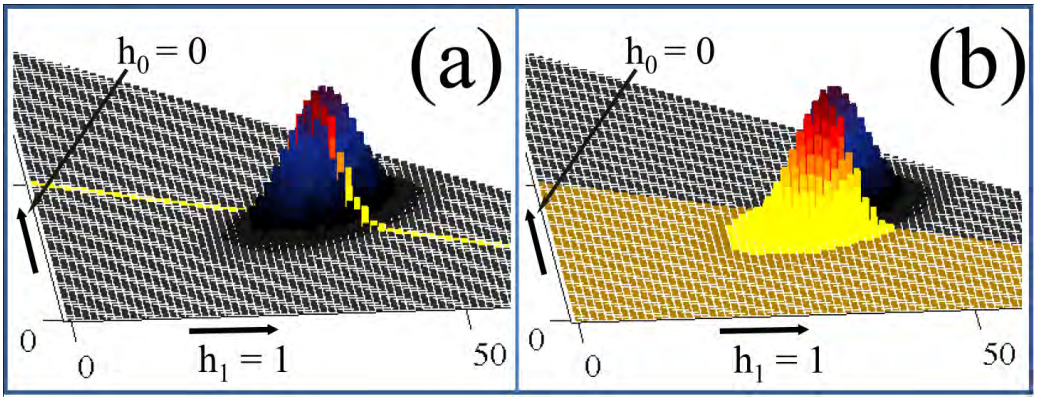


Figure 4.3: Histogram space for multinomial distributions with (a) equality and (b) inequality constraints.

As in the previous section, the multinomial distribution can be approximated by the multivariate normal distribution. Suppose that each rv may have three values with probability p_0, p_1 and p_2 ($p_0{+}p_1{+}p_2 = 1$), respectively. Then the multinomial distribution of h_0, h_1 and h_2 chosen from n rvs is

$$\binom{n}{h_0\, h_1\, h_2} \cdot p_0^{h_0} \cdot p_1^{h_1} \cdot p_2^{h_2} = \frac{n!}{h_0!h_1!h_2!} \cdot p_0^{h_0} \cdot p_1^{h_1} \cdot p_2^{h_2}.$$

The corresponding bivariate (i.e. (3-1) multivariate) normal distribution of $\mathbb{X}$ =

$[h_0\, h_1]$ chosen from n rvs is as follows (Note that $h_2 = n - h_1 - h_2$),

$$\frac{1}{(2\pi)^{2/2}|\Sigma|^{1/2}} \cdot exp\left(-\frac{1}{2}(\mathbb{X} - \mu)\Sigma^{-1}(\mathbb{X} - \mu)'\right),$$

when the μ and Σ are

$$\mu = [np_0\, np_1],\ \Sigma = \begin{pmatrix} np_0(1-p_0) & np_1p_2 \\ np_2p_1 & np_2(1-p_2) \end{pmatrix}.$$

Analytical Solution for Operators with a Single Linear Constraint

As in the previous section, one sets p_0, p_1 and p_2 as 0.35, 0.35 and 0.3 respectively and y as 100. Any operator with a single linear constraint (e.g. **AFM**, **AFM**[$\leq$] and **AFM**[$\geq$] on *SUM*, and **AFM**[$\leq$] and **AFM**[$\geq$] on *MEDIAN*) allows an analytical solution because there is a linear transformation from $\mathbb{X} = [h_0\, h_1]$ to y. Consider the following linear transform $y = 0{\cdot}h_0 + 1{\cdot}h_1 + 2{\cdot}h_2 = 200 - 2{\cdot}h_0 - h_1$. When one represents the transform as $y = A\mathbb{X} + B$, the new distribution of y is given by the 1-D Normal distribution:

$$\frac{1}{\sqrt{2\pi\Sigma_y}} \cdot exp\left(-\frac{(y-\mu_y)^2}{2\Sigma_y}\right),$$

where $\mu_y = A\mu + B$ and $\Sigma_y = A\Sigma A^T$ are scalars. From the transformation the solution of **AFM** for y=100 can be calculated in the following way:

$$\frac{1}{\sqrt{2\pi\Sigma_y}} \int_{y=100-0.5}^{100+0.5} exp\left(-\frac{(y-\mu_y)^2}{2\Sigma_y}\right) dy.$$

The solutions of **AFM**[$\leq$] and **AFM**[$\geq$] for y=100 can be calculated in similar ways.

Sampling for Remaining Operators

In general, integration of a multivariate truncated normal does not allow an analytical solution. Fortunately, efficient **Gibbs sampling** methods (e.g. [Geweke, 1991; Damien and Walker, 2001]) are applicable to the truncated normal in straightforward ways, even with several linear constraints. This immediately feeds to an approximation with time complexity not depending on n, the number of rvs.

4.4 Aggregate Factor with Multiple Atoms

This section now considers a generalized situation. Previous sections assume that all rvs in a relational atom have the same distribution. Here, it deals with the issue of aggregating J distinct groups of random variables, each represented by a relational atom X_j with n_j groundings and a distinct potential $\phi_{\mathbf{x_j}}$, for $1 \leq j \leq J$.

$$y = \bigotimes_{\substack{1<j<J \\ 1<i<n_j}} x_{j,i}.$$

This problem, **AFM-M**, is an extension of the **AFM**. The **AFM-M** is to calculate a marginal

$$\sum_{x_{1,1},\cdots,x_{J,n_J}} \phi_{\otimes}(y, x_{1,1}, \cdots, x_{J,n_J}) \prod_{j=1}^{J} \prod_{1 \leq i \leq n_j} \phi_{\mathbf{x_j}}(x_{j,i}).$$

One approach is to compute an aggregate y_j^0 per atom j, and then combine each pair y_j^i and y_{j+1}^i into $y_{\lfloor j/2 \rfloor}^{i+1}$ until they are all aggregated. This will have complexity $O(J \log J)$ but works only for associative operators. For non-associative operators, one may need to calculate the marginal for each X_j independently:

$$\sum_{h^1,\cdots,h^J} \phi_{\otimes \mathbf{h}}(y, \mathbf{h}) \left(\binom{n_1}{h_1^1} p_{1,0}^{h_0^1} p_{1,1}^{h_1^1} \cdots \binom{n_J}{h_1^J} p_{J,0}^{h_0^J} p_{J,1}^{h_1^J} \right),$$

where $p_{j,0}$ and $p_{j,1}$ are the normalization of $\phi_{\mathbf{x_j}}(0)$ and $\phi_{\mathbf{x_j}}(1)$; h^j is a histogram for atom j, and $\mathbf{h}$ is the combined histogram. The complexity of this approach is $O(exp(J))$.

Another approach is to make use of the representation of the aggregation operator as a set of linear constraints (Table 4.1). Note that h^j is approximately Normal when n_j is large, and h^i and h^j are independent when $i \neq j$. Thus, the all-group histogram vector $\mathbf{h}$ is also approximately Normal distributed because it is the Normal sum ($\mathbf{h}_i = \sum_j h_i^j$).

Any linear constraint in Table 4.1 can be re-expressed as a linear constraint using elements of $\mathbf{h}$, and the multinomial-Normal approximation can be used to yield a similar approximate solution in time constant n, the total number of rvs.

For example, for binary random variables, the Normal approximation of the all-group histogram is:

$$N\left(\sum_{j=1}^{J} n_j p_{j,1}, \sum_{j=1}^{J} n_j p_{j,1} p_{j,0}\right).$$

This way, the time complexity is only $O(J)$ instead of $O(J \log J)$ (or $O(exp(J))$ for non-associative operators).

4.5 Error Analysis

Here, this section discusses error bounds for the multinomial-Normal approximations. In general, the Berry-Esseen theorem [Esseen, 1942] gives an upper bound on the error. Suppose that $\phi_{\mathbf{y}}(y)$ and $\widetilde{\phi}_{\mathbf{y}}(y)$ represent the probability mass of a binomial distribution and density of its normal approximation, respectively. Furthermore, one may represent the cumulative probabilties as $\Phi_{\mathbf{y}}(y)$ and $\widetilde{\Phi}_{\mathbf{y}}(y)$[6]. Then, given any y, the error between the two cumulative probabilities is bounded [Esseen, 1942]:

[6]That is, $\Phi_{\mathbf{y}}(y) = \sum_{i=0}^{y} \phi_{\mathbf{y}}(i)$, and $\widetilde{\Phi}_{\mathbf{y}}(y) = \int_{t=-\infty}^{y} \widetilde{\phi}_{\mathbf{y}}(t)\, dt$.

$$\left|\Phi_{\mathbf{y}}(y) - \widetilde{\Phi}_{\mathbf{y}}(y)\right| < c \cdot \frac{p^2 + (1-p)^2}{\sqrt{np(1-p)}},$$

where c is a small (< 1) constant. Thus, the asymptotic error bound is $O(1/\sqrt{n})$, and this extends to probability on any interval.

For k-valued multinomials, suppose that $\Phi_{\mathbf{Y}}(A)$ and $\widetilde{\Phi}_{\mathbf{Y}}(A)$ represent the probability of a multinomial distribution and its multivariate normal approximation over a measurable convex set A in R^k. Then, the approximation error is bounded [Gotze, 1991]:

$$sup_A \left|\Phi_{\mathbf{Y}}(A) - \widetilde{\Phi}_{\mathbf{Y}}(A)\right| < c \cdot \frac{k}{\sqrt{n}},$$

where c depends only on the multinomial parameters and not on n. In our problem, A is determined by linear constraints, hence is convex. Thus, the asymptotic error bound is $O(k/\sqrt{n})$.

4.6 Experimental Results

This section provides experimental results on the example in Figure 1 (which uses the *MODE* aggregate function) which give us an insight on when to use the approximate algorithm instead of the generally applicable exact algorithm based on Counting Formulas (the logarithmic method in [Kisynski and Poole, 2009] does not apply to *MODE*).

One may compute the utility of any of the methods tested, approximations or exact inference alike, in the following manner. One assumes a typical application in which the utility of an error is an inverse quadratic function $U(err) = 1 - err^2$. The utility of a method obtaining error err is normalized by the time t it takes to run, so $U(err, t) = U(err)/t$. For sampling methods, t is the time to convergence. Finally, it plots the *ratio* between the utility of our methods and the utility of the exact inference method.

Therefore, a method is advantageous over the exact inference method when this

ratio is greater than 1.

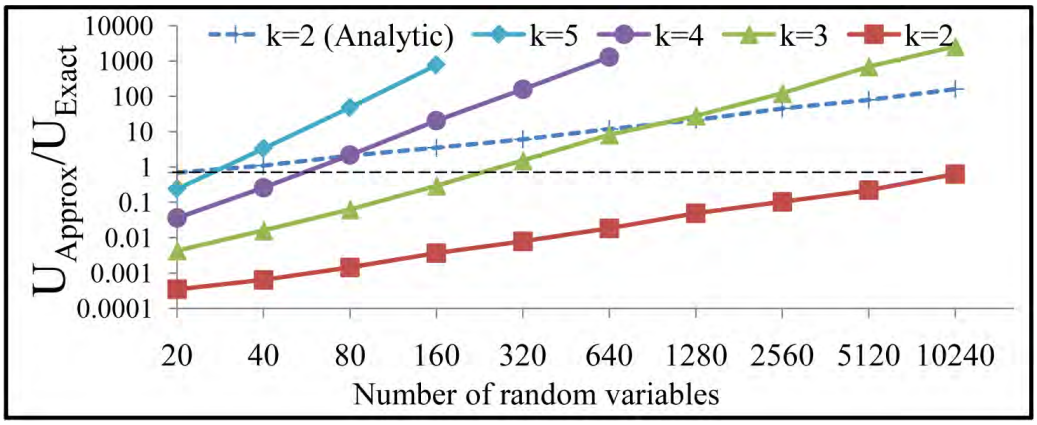


Figure 4.4: Ratios of utilities of approximate algorithms and exact method (histogram based counting).

An experiments runs on the approximations and the exact inference algorithm for the model in Figure 4.1. For $k = 2$, it runs both the analytical and the sampling method. Given k and n, it randomly chooses the potentials, and records the error and the convergence time. Then, it average them over 100 trials to calculate the utility, U_{Approx}.

As shown in Figure 4.4, the suggested approximate algorithm has much higher utility than the exact method for larger k and n. However, when $k = 2$ (binary variables), the exact method has higher utility than sampling for relatively large n (e.g. $n = 10240$). In this case, one can use the efficient analytic integration which applies for $k = 2$. I also show in Figure 4.5 how the error decreases for different values of k and n.

In addition, the results observed that the convergence time stays flat for various k and n. However, the error of sampling method is noticeable for small n. For example, when $k = 4$, the error is 3.07% with $n = 40$ and 1.82% with $n = 80$. For larger n, this issue is resolved. The error becomes less than 1% when $n = 320$ and negligible when $n > 5120$. These observations are consistent for various k from 2 to 6.

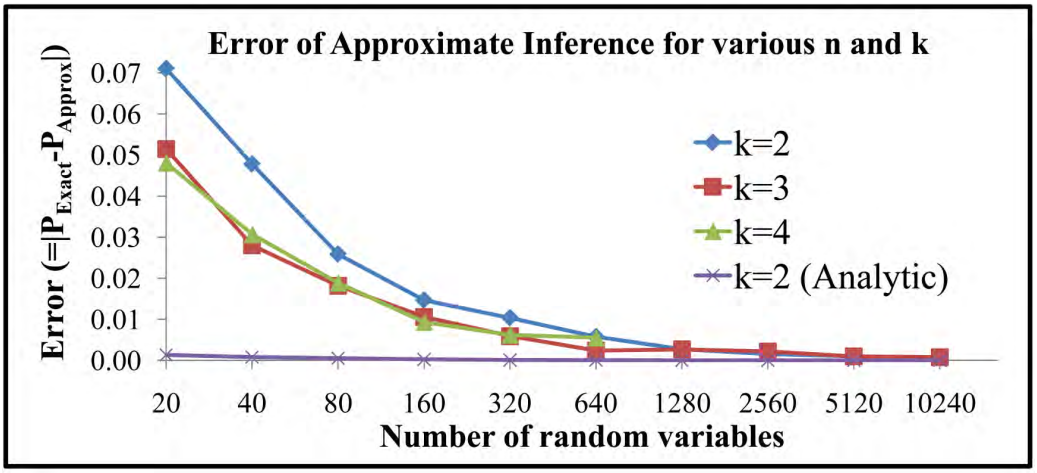


Figure 4.5: Error curves for different values of k and n.

4.7 Conclusion

Processing aggregate parfactors efficiently is an important problem since they involve functions commonly used in writing models. Our contribution adds efficient exact methods for the binary case k=2, as well as efficient approximations for the cases in which the sets of aggregated variables are large, which is precisely the situation in which one may be more likely to use aggregate factors in the first place. It will therefore be an important part of practical applications of relational graphical models.

CHAPTER 5

LIFTED VARIATIONAL INFERENCE

Hybrid continuous-discrete models naturally represent many real-world applications in robotics, finance, and environmental engineering. Inference with large-scale hybrid models is challenging because structures deteriorate rapidly during inference with observations. The main contribution of this chapter is the efficient relational variational inference algorithm that factors large-scale probability models into simpler variational models, composed of mixtures of iid (Bernoulli) random variables. The algorithm takes probability relational models of large-scale hybrid systems and converts them to a close to optimal variational approximation. Then, it efficiently calculates the marginal probabilities on the variational models by using latent lifted variable elimination or lifted stochastic sampling. This inference is unique because it maintains a relational structure upon individual observations and during inference steps.

5.1 Introduction

Many real-world systems can be described using continuous and discrete variables with relations among them. Such examples include measurements in environmental sensor networks, localizations in robotics, and economic forecasting in finance. In such large systems, efficient and precise inference is essential. As an example from environmental engineering, an inference algorithm can predict a posterior of unobserved groundwater levels and contamination levels at different locations, and making such an inference precisely is critical to decision makers [Xu *et al.*, 2012].

Real-world systems have large numbers of variables including both discrete and continuous. Probabilistic first order languages, e.g. [Bacchus, 1990; Halpern, 1990; Ng and Subrahmanian, 1992; Pfeffer *et al.*, 1999; Friedman *et al.*, 1999; Poole, 2003; Richardson and Domingos, 2006], describe probability distributions at a relational level with the purpose of capturing the structure of larger models. A key challenge of inference procedures with the languages is that they often result in intermediate density functions involving many random variables and complex relationship among them.

Lifted inference presently can address discrete models and continuous models, but not hybrid ones. For (d-valued) discrete variables, lifted inference can take an advantage of the insight which groups equivalent models into a histogram representation with an order of $poly(d)$ entries [de Salvo Braz *et al.*, 2005; Milch and Russell, 2006; Jha *et al.*, 2010] (instead of $exp(d)$ entries in traditional *ground* models). For Gaussian potentials, lifted inference can use an insight which enables maintaining compact covariance matrices during (and after) inference, e.g. [Choi *et al.*, 2010a].

Unfortunately, these principles are not applicable to general (non-Gaussian) hybrid models because the histogram is not applicable to continuous domains without discretizations, and the covariance matrix is a special structure for Gaussians. Thus, existing variational methods, e.g. NP-BLOG [Carbonetto *et al.*, 2005] and Latent Tree Models [Zhang, 2002; Choi *et al.*, 2011d], focus only on discrete or Gaussian models.

In this chapter, we present pragmatic algorithms based on a new insight (relational variational-inference lemmas) which accurately factors densities of relational models into mixtures of iid random variables. These lemmas enable us to build a variational approximation algorithm, which takes large-scale graphical models with hybrid variables and finds close to optimal relational variational models. Then, our inference algorithms, the variable elimination and the stochastic sampling, efficiently solve marginal inference problems on the variational models. We show that the algorithm gives a better solution than previous ones.

This chapter is organized as follows. Section 5.2 defines **Relational Hybrid Models**. Section 5.3 presents a theoretical background, de Finetti's theorem. Section 5.4 overviews our **Lifted Relational Variational Inference** algorithm. Section 5.5 and 5.6 respectively elucidate the learning and inference algorithms. Section 5.7 presents our theoretical contributions, **relational-variational lemmas**. Section 5.8 compares our method with existing inference methods. Section 5.9 shows experimental results, followed by the conclusion and future work in Section 5.10.

5.2 Relational Hybrid Models (RHMs)

A **factor** $f = (A_f, \phi_f)$ is a pair, composed of a tuple of random variables (**rv**s) A_f and a potential function ϕ_f. Here, ϕ_f is an unnormalized probability density from the range of A_f to the nonnegative real numbers. The range of a rv is discrete or continuous, i.e., hybrid domains. Given a **valuation v** of rvs, the **potential** of f on $\mathbf{v}$ is $w_f(\mathbf{v}) = \phi_f(A_f)$.

We define **parameterized (indexed) rvs** by using **predicates** those are functions mapping parameter values to rvs. A **relational atom** (or just **atom**) denotes a parametrized rv with free parameter variable(s). For example, an atom $\mathbf{X}(\mathrm{a})$ can be mapped to one of n rvs $\{X(a_1), \cdots, X(a_n)\}$ when the free parameter variable a is **substituted** by a value a_i.

A **parfactor** $g = [PV, A_g, \phi_g]$ is a tuple composed of parameter variables PV, a tuple of relational atoms A_g and a potential function ϕ_g. A substitution θ is an assignment to PV, and $A_g\theta$ the relational atom (possibly ground) resulting from replacing the logical variables by their values in θ. $gr(g)$ is a set of factors derived from the parfactor g by substitutions.

For example, a RHM can include the following parfactor:

$$[\ \underbrace{(\mathsf{a},\mathsf{b})}_{\text{Parameter variables}}\ ,\underbrace{(\mathbf{X}(\mathsf{a}),\mathbf{Y}(\mathsf{b}))}_{\text{Relational atoms}},\underbrace{\mathcal{N}(\mathbf{X}(\mathsf{a})-\mathbf{Y}(\mathsf{b});\mu,\sigma^2)}_{\text{A potential (linear Gaussian)}}]. \tag{5.1}$$

The domains of the parameter variables (a and b) can be $\{a_1,\ldots,a_n\}$ and $\{b_1,\ldots,b_m\}$. Thus, any substitution (e.g. $\mathsf{a}=a_i,\mathsf{b}=b_j$) let two rvs (e.g. $\mathbf{X}(\mathsf{a}_\mathsf{i})$, $\mathbf{Y}(\mathsf{b}_\mathsf{j})$) holds the linear Gaussian relationship.

A **Relational Hybrid Model (RHM)** is a compact, or intensional, representation of graphical models with discrete and continuous rvs. An RHM is composed of a **domain**, the set of possible parameter values, and a set of parfactors $\mathbf{G}$. The joint probability of an RHM G on a valuation v of rvs is as follows:

$$\frac{1}{z}\prod_{g\in\mathbf{G}}\prod_{f\in gr(g)} w_f(\mathbf{v})$$

where z is the normalizing constant.

This representation seems rather straightforward. However, inference procedures often result in complex models. For example, eliminating $X(a_1)$ in Equation (5.1) makes all other rvs fully connected. An important property in RHMs is that the ground rvs mapped from a relational atom are **exchangeable**, defined as follows:

Definition 1 (Exchangeable Random Variables) *A sequence $X(a_1),\cdots,X(a_n)$ of rvs is* **exchangeable***, when for any finite permutation $\pi()$ of the indices the joint probability distribution of the permuted sequence $X(a_{\pi(1)}),\cdots,X(a_{\pi(n)})$ is the same as the joint probability distribution of the original sequence.*

Note that RHMs may include atoms with non-exchangeable rvs.[1] In this case, our variational algorithm grounds, or shatters, any atom including non-exchangeable

[1]Lifted inference for non-exchangeable rvs is out of scope of this chapter. Instead, see [Jha *et al.*, 2010; Apsel and Brafman, 2011].

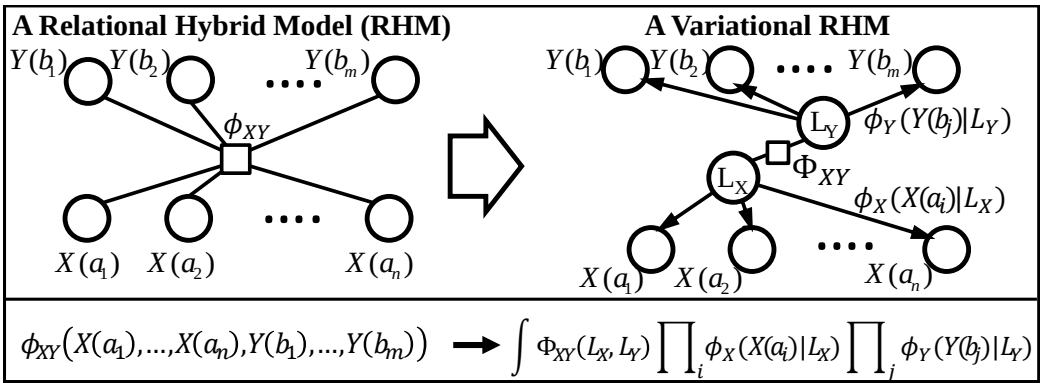


Figure 5.1: An illustration of factoring a potential $\phi_{XY}(\mathbf{X}^n, \mathbf{Y}^m)$. Our algorithm converts a RHM (left) into a variational (or factored) RHM (right) where the probability is represented by only two latent variables L_X and L_Y.

rvs. That is, the atom becomes a set of propositional rvs. The detail conditions to determine exchangeable rvs, see [Poole, 2003; Milch *et al.*, 2005; Carbonetto *et al.*, 2005; de Salvo Braz *et al.*, 2005].

For convenience, we use $\mathbf{X}^n$ to simplify the set of n rvs, which are mapped from a relational atom $X(\mathrm{a})$. That is, $\mathbf{X}^n = \{X(a_1), \cdots, X(a_n)\}$. The joint probability of the rvs, which are mapped from two atoms $\mathbf{X}(\mathrm{a})$ and $\mathbf{Y}(\mathrm{b})$, can be represented as follows: $P(\mathbf{X}^n, \mathbf{Y}^m) = P(X(a_1), \cdots, X(a_n), Y(b_1), \cdots, Y(b_m))$.

Potentials with a large number of rvs in RHMs introduce several difficulties in representation, learning and inference. To address these difficulties, we propose a model-factorization based on a variational method, based on de Finetti's theorem [de Finetti, 1931], as shown in Figure 5.1.

5.3 Background

De Finetti-Hewitt-Savage's Theorem: Before introducing our result, our algorithm and a new error analysis, we review de Finetti's theorem [de Finetti, 1931] which shows that $P(\mathbf{X}^n)$ any probability distribution of an infinite number of binary exchangeable rvs can be represented by $P_{Miid}(\mathbf{X}^n)$[2] a mixture of independent

[2]Here, *Miid* refers to a mixture of iid rvs.

and identically distributed (**iid**) Bernoulli rvs with a parameter θ:

$$\lim_{n\to\infty} P(\mathbf{X}^n) = \int_0^1 \theta^{t_n}(1-\theta)^{n-t_n}\Phi_X(\theta)\,\mathrm{d}\theta = P_{Miid}(\mathbf{X}^n),$$

when $t_n = \sum_i X(a_i)$. This observation is extended to multi-valued and continuous rvs by [Hewitt and Savage, 1955].

$$\lim_{n\to\infty} P(\mathbf{X}^n) = \int \Phi_X(L_X) \prod_{i=1}^{n} \phi_X(X(a_i)|L_X)\,\mathrm{d}L_X = P_{Miid}(\mathbf{X}^n), \tag{5.2}$$

where L_X is a new **latent variable (LV)** which chooses a distribution $\phi_X(\mathbf{X}(a)|L_X)$ of the iid rvs.[3] When $\phi(\mathbf{X}^n)$ is given, the variational form is represented as $\phi_{Miid}(\mathbf{X}^n)$. The number of parameters, e.g. entries in the conditional distribution table (CDT), of the potential $\phi_{Miid}()$ is substantially reduced by this factorization. When the variational models is applied to two sets of exchangeable rvs, as shown in in Figure 5.1, the variational model (the right hand side) requires parameters of $\phi_X(X(a_i)|L_X)$, $\phi_Y(Y(b_j)|L_X)$ and $\Phi_X(L_X)$ only.

The exchangeability theorems are exact only for a single set of infinite, exchangeable rvs. For multiple sets of finite, exchangeable rvs, it is natural to analyze the variational error which we present in Section 5.7. Before doing that, we present our learning and inference algorithms.

5.4 Algorithm: Lifted Inference with RHMs

This section outlines our efficient variational inference algorithm for RHMs, ***Lifted Relational Variational Inference*** **(LRVI)**. The LRVI is composed of two main subroutines: a variational approximation, ***Find-Variational-RHM***, and a variable elimination, ***Latent-Variational-Elimination*** which can be replaced by ***Lifted-MCMC***.

[3]The right hand side of Equation (5.2) is $\int \prod_{i=1}^{n} \phi_X(X(a_i)|L_X) \cdot \Phi_X(\mathrm{d}L_X)$. It is possible to replace $\Phi_X(\mathrm{d}L_X)$ with $\Phi(L_X)\,\mathrm{d}L_X$, when Φ_X has a distribution. Here, we only consider such Φ_X.

PROCEDURE **LRVI**(G,Q,O)
An *RHM* (a set of parfactors), G; a query (a set of relational atoms), Q; observations, O.
1. // (One-time) Variational Learning
2. **If** ($G \notin$ {Variational RHMs})
   (a) $G \leftarrow$ **Find-Variational-RHM**(G)
3. // Main Inference Routine
4. $P(Q) \leftarrow$ **Latent-Variable-Elimination** (G, Q, O)
5. **Return** $P(Q)$

Figure 5.2: Algorithm Lifted Relational Variational Inference (LRVI).

The LRVI receives an RHM G, a query Q and observations O as inputs. It outputs the conditional probability, $P(Q|O)$. In the routine, it examines that each potential in G is the variational form, a mixture of product of iid rvs. If not, it calls *Find-Variational-RHM(G)* and receives a variational RHM G_{Miid}. The variational RHM is calculated once, and reused next time. With the G_{Miid}, *Latent-Variable-Elimination(G_{Miid}, Q, O)* solves the inference problem $P(Q|O)$. This is done by the variable elimination which iteratively eliminates non-query atoms.

PROCEDURE **Find-Variational-RHM**(G)
An *RHM* (a set of parfactors), G.
1. **For** $g = (L, A, \phi) \in G$
   (a) **If** (A has no continuous atom)
   $\phi_{Miid} \leftarrow$ **Lifting-Discrete(ϕ)** (Section 5.5.1)
   Else $\phi_{Miid} \leftarrow$ **Lifting-Continuous(ϕ)** (Section 5.5.2)
   (b) $G_{Miid} \leftarrow G_{Miid} \bigcup \{(L, A, \phi_{Miid})\}$
2. **Return** G_{Miid}

Figure 5.3: Algorithm Find-Variational-RHM (Section 5.5).

Find-Variational-RHM(G) converts the potential ϕ in each parfactor into a variational potential ϕ_{Miid}, a mixture of iid rvs as shown in Equation (5.2). For potentials with only discrete atoms, it calls *Lifting-Discrete(ϕ)* and receives a variational potential ϕ_{Miid}. For potentials including any continuous atom, it calls *Lifting-Continuous(ϕ)*. After iterating and converting all parfactors in G, a variational RHM G_{Miid} is returned. Section 5.5 explains the details.

Latent-Variable-Elimination(G_{Miid}, Q, O) first handles observations by calling

PROCEDURE **Latent-Variable-Elimination**(G,Q,O)
A variational *RHM*, G_{Miid}; a query, Q; observations O.
1. $G_{Miid} \leftarrow$ Update-Obs(G_{Miid}, O) (for observations)
2. $\mathbf{A} \leftarrow$ a set of atoms in G_{Miid}
3. $\mathbf{\Phi} \leftarrow \{\phi_g | g \in G_{Miid}\}$
4. **For** $X \in \mathbf{A} \setminus Q$
   (a) $\Phi_X \leftarrow \{\phi \in \mathbf{\Phi} | X \text{ is argument of } \phi\}$
   (b) **If** (X is discrete) $\phi' \leftarrow$ **Inference-Discrete**(Φ_X) (Section 5.6.1)
   Else $\phi' \leftarrow$ **Inference-Continuous**(Φ_X) (Section 5.6.2)
   (c) $\mathbf{\Phi} \leftarrow (\mathbf{\Phi} \setminus \phi_X) \bigcup \{\phi'\}$
5. **Return** $\mathbf{\Phi}$

Figure 5.4: Algorithm Latent-Variable-Elimination (Section 5.6).

Update-Obs(G_{Miid}, O) to update the potentials of LVs based on O. The intuition of *Update-Obs()* is that each rv is conditionally independent given LVs like the Naive Bayes models [John and Langley, 1995]. Thus, it is possible to build a simple update algorithm which maintains the relational structure upon individual observations.[4] It iteratively eliminates all latent variables except the query without referring ground variables. Section 5.6 explains the procedures in detail.

5.5 Variational Learning in RHMs

This section elaborates a learning algorithm which converts each potential in an RHM into a variational potential. Here, the key procedure is to extract the probability on LVs (e.g. $\Phi_X(L_X)$). When an input potential satisfies a condition, ∞-extendible (explained in Section 5.7), the cumulative distribution function (cdf) on LVs can be derived analytically and exactly [Diaconis, 1977].

It is also known that discrete potentials[5] and some Gaussian potentials (e.g. pairwise Gaussian [Choi *et al.*, 2010a] and Gaussian processes [Chu *et al.*, 2006; Xu *et al.*, 2009]) allow such derivations. Unfortunately, it is hard to use such derivations in general because many real-world potentials are neither ∞-extendible

[4]Most existing algorithms degenerate relational models upon observations. For details, see Split [Poole, 2003] or Shatter [de Salvo Braz *et al.*, 2005].

[5]Section 5.8 includes some comparison with existing lifted inference for discrete models

nor Gaussian. Here, we present rather intuitive variational discrete models first, and then focus on continuous ones.

5.5.1 Lifting Discrete Potentials

For discrete potentials, we need to find the probability density $\Phi_X(L_X)$ over the iid (Bernoulli) rvs where L_X is the Bernoulli parameter. To represent an input potential $\phi(\mathbf{X}^n)$ compactly, we group equivalent value assignments by the **value-histogram representation** [de Salvo Braz *et al.*, 2005; Milch *et al.*, 2008], $\phi(\mathbf{X}^n) = \phi_{\mathbf{h}}(h_X)$.[6] Equation (5.2) with discrete rvs is formulated as follows:

$$\begin{aligned}
&\arg\max_{\Phi_X(L_X)} \left\| \phi_X(\mathbf{X}^n) - \int \Phi_X(L_X) \prod_{i=1}^{n} \phi_X(X(a_i)|L_X)\, dL_X \right\| \\
&\quad = \arg\max_{\Phi_X(L_X)} \left\| \phi_{\mathbf{h}}(h_X) - \int \Phi_X(L_X) \cdot f_{\mathcal{B}/\mathcal{M}}(h_X; n, L_X)\, \mathrm{d}L_X \right\| \\
&\quad \approx \arg\max_{\mathbf{w}, \mathbf{l_X}} \left\| \phi_{\mathbf{h}}(h_X) - \sum_{u=1}^{k} w_u \cdot f_{\mathcal{B}/\mathcal{M}}(h_X; n, l_{X,u}) \right\|, \qquad (5.3)
\end{aligned}$$

where $f_{\mathcal{B}/\mathcal{M}}(h_X; n, L_X)$ is a binomial (or multinomial) pdf; $\mathbf{w} = (w_1, \cdots, w_k)$ is a k-dimensional weight vector such that $\sum_{u=1}^{k} w_u = 1$; and $\mathbf{l_X} = (l_{X,1}, \cdots, l_{X,k})$ is a vector of k values chosen from the latent variable L_X.

For **binary** rvs $\mathbf{X}^n$, the iid potential $\phi_X(X(a_i)|l_{X,u})$ is the Bernoulli distribution with a parameter l_X (i.e. $P(X(a_i)) = l_{X,u}$). When equivalent models in $\mathbf{X}^n$ are grouped into the histogram h_X, the product of n Bernoulli distributions $\prod_i \phi_X(X(a_i)|L_X)$ along with number of equivalent models $\binom{n}{h_X}$ forms a binomial distribution because $f_{\mathcal{B}}(h_X; n, L_X) = \binom{n}{h_X} \prod_i \phi_X(X(a_i)|L_X)$. That is, the problem is reduced to learn a mixture of binomial distributions where w_u is a weight for each binomial $f_{\mathcal{B}}(h_X; n, l_{X,u})$.

For **multi-valued** rvs $\mathbf{X}^n$, the iid potential $\phi_X(X(a_i)|l_{X,u})$ is the Categorical distribution, i.e. multi-valued Bernoulli. Thus, this problem is reduced to learn a

[6] h_X is a vector with $h_{X_v} = |\{i : X(a_i) = v\}|$.

mixture of multinomial distributions $f_{\mathcal{M}}$:

$$\arg\max_{\mathbf{w},\mathbf{l_X}} \left\| \phi_{\mathbf{h}}(h_X) - \sum_{u=1}^{k} w_u \cdot f_{\mathcal{M}}(h_X; n, l_{X,u}) \right\|. \tag{5.4}$$

For potentials with two or more atoms, it can be formulated as follows:

$$\arg\max_{\mathbf{w},\mathbf{l_X},\mathbf{l_Y}} \left\| \phi_{\mathbf{h}}(h_X, h_Y) - \sum_{u=1}^{k} w_u f_{\mathcal{B}/\mathcal{M}}(h_X; n, l_{X,u}) f_{\mathcal{B}/\mathcal{M}}(h_Y; m, l_{Y,u}) \right\|, \tag{5.5}$$

where $\mathbf{l_Y}$ is a k-dimensional vector, $(l_{Y,1}, \cdots, l_{Y,k})$ and $f_{\mathcal{B}/\mathcal{M}}$ is the binomial or multinomial distribution depends of the domain (binary or multi-valued) of rvs.

We learn a mixture of binomials (or multinomials), i.e. $(\mathbf{w}, \mathbf{l_X})$, in Equation (5.3), (5.4) and (5.5), from the original potential ϕ using an incremental EM algorithm.[7] Because the k is not known or given, the incremental EM algorithm increases k up to n until the variational error converges.[8] The computational complexity of the EM algorithm for n binary, exchangeable rvs is bounded by $O(n \cdot \frac{n^2}{2})$. The EM algorithm incrementally increases k from 1 to n. In each EM step, k components visit n histogram entries $O(kn)$.

5.5.2 Lifting Continuous and Hybrid Potentials

For a potential $\phi_X(\mathbf{X}^n)$ with continuous rvs, we use a mixture of non-parametric densities to represent variational potentials. Here, we generate samples from the input potential $\phi_X(\mathbf{X}^n)$, then learn parameters for the mixture of non-parametric densities.

[7] EM algorithms are common to learn parameters for mixture models [Xu and Jordan, 1995; Dasgupta, 1999; Rasmussen and Ghahramani, 2001].

[8] The value-histogram is an exact representation with $poly(n)$ entries. Thus, it is not reasonable to build an approximate variational model with more than $poly(n)$ entries. For conditions and examples, we explain details in Section 5.7 and Figure 5.5.

Equation (5.2) is used to formulate the learning problem as follows:

$$\arg\max_{\Phi_X(L_X)} \left\| \phi_n(\mathbf{X}) - \int \Phi_X(L_X) \cdot \prod_{i=1}^{n} \hat{f}_{L_X}(X(a_i))\, \mathrm{d}L_X \right\| \tag{5.6}$$

$$\approx \arg\max_{\mathbf{w}, \hat{\mathbf{f}}_{\mathbf{l_X}}} \left\| \phi_n(\mathbf{X}) - \sum_{u=1}^{k} w_u \cdot \prod_{i=1}^{n} \hat{f}_{l_{X,u}}(X(a_i)) \right\|, \tag{5.7}$$

where $\hat{f}_{L_X}$ refers a probability distribution (possibly non-parametric). To solve the optimization problem, we generate N samples $\mathbf{v_1}, \cdots, \mathbf{v_N}$ from the input potential $\phi(\mathbf{X}^n)$ where $\mathbf{v_t} = (v_{t,1}, \cdots, v_{t,n})$, values of n rvs. Then, we solve the following maximum likelihood estimation (MLE) problem:

$$\arg\max_{\mathbf{w}, \hat{\mathbf{f}}_{\mathbf{l_X}}} \sum_{t=1}^{N} \log \left(\sum_{u=1}^{k} w_u \cdot \prod_{i=1}^{n} \hat{f}_{l_{X,u}}(v_{t,i}) \right),$$

where we denote the kernel density estimator by $\hat{f}_{l_{X,u}}(x) = \frac{1}{S\sigma^2} \sum_{i=1}^{S} K\left(\frac{x-\mu_i}{\sigma^2}\right)$ where $(\mu_1, \cdots, \mu_S)$ are S data points that underlie the density, and σ^2 is a parameter. For simplicity, we use the Gaussian Kernel, $K(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2}$.

It is interesting to note that the kernel density estimator is analogous to the value-histogram for discrete rvs in a sense that frequently observed regions (or bins) have a higher probability. This new insight enables us to represent continuous models compactly .

For potentials with two or more atoms, the approach can be formulated as follows:

$$\arg\max_{\mathbf{w}, \hat{\mathbf{f}}_{\mathbf{l_X}}, \hat{\mathbf{f}}_{\mathbf{l_Y}}} \sum_{t=1}^{N} \log \left(\sum_{u=1}^{k} w_l \cdot \prod_{i=1}^{n} \hat{f}_{l_{X,u}}(v_{X_{t,i}}) \cdot \prod_{j=1}^{m} \hat{f}_{l_{Y,u}}(v_{Y_{t,j}}) \right),$$

where $v_{X_{t,i}}$ is the value of i^{th} rv of X in the t^{th} sample, and $v_{Y_{t,j}}$ is the value of j^{th} rv of Y.

This MLE problem is also solved by an EM algorithm. N samples are used to build k densities in the maximization (M) step, and the likelihood of each sample

is calculated in the expectation (E) step.

5.6 Lifted Inference with Variational RHMs

In this section we build on the result of previous sections, variational RHMs, and present lifted inference algorithms that utilize the learned variational models to speed up relational inference. The algorithms are designated lifted variational inference because they solve inference problems without referring all rvs.

The *Latent-Variable-Elimination()* marginalizes relational atoms with the following steps: (i) choosing an atom; (ii) finding all potentials including the atom and making a product of them; (iii) marginalizing the atom; and (iv) repeating the steps until only query atoms are left. We demonstrate the key step (iii) with an example of two variational potentials, $\phi_{Miid}(\mathbf{X}^n, \mathbf{Y}^m)$ and $\phi'_{Miid}(\mathbf{Y}^m)$ as shown in Equation (5.5).

5.6.1 Inference with Discrete Variables

The key intuition is that the variational form is maintained after eliminating an atom. We demonstrate the intuition by an example. The marginal probability of the LV L_X is calculated by eliminating (or summing) $\mathbf{Y}^m$ out: $\sum_{h_y} \phi_{\mathbf{h}}(h_x, h_y) \cdot \phi'_{\mathbf{h}}(h_y)$

$$
\begin{aligned}
&\approx \sum_{h_y} \sum_{u=1}^{k} w_u f_{\mathcal{B}}(h_x; n, l_{X,u}) f_{\mathcal{B}}(h_y; m, l_{Y,u}) \sum_{u'=1}^{k'} w_{u'} f_{\mathcal{B}}(h_y; m, l_{Y,u'}) \\
&= \sum_{u=1}^{k} \sum_{u'=1}^{k'} w_u w_{u'} \left(\sum_{h_y} f_{\mathcal{B}}(h_y; m, l_{Y,u}) f_{\mathcal{B}}(h_y; m, l_{Y,u'}) \right) f_{\mathcal{B}}(h_x; n, l_{X,u}) \\
&\approx \sum_{u=1}^{k} \sum_{u'=1}^{k'} w_u w_{u'} \left(\int f_{\mathcal{N}}(h_y; \mu_u, \sigma_u^2) f_{\mathcal{N}}(h_y; \mu_{u'}, \sigma_{u'}^2) \, \mathrm{d}h_y \right) f_{\mathcal{B}}(h_x; n, l_{X,u}) \\
&= \sum_{u=1}^{k} w_{\overline{Y},u} \cdot f_{\mathcal{B}}(h_x; n, l_{X,u}) = \phi''(\mathbf{X}^n)
\end{aligned}
\tag{5.8}
$$

when $\sum_{l=1}^{k} w_{\overline{Y},l}$=1 and $f_{\mathcal{N}}(h_y; \mu_u, \sigma_u^2)$ is the Normal approximation to binomial such that $\mu_u(=ml_{Y,u})$ and $\sigma_u^2(=ml_{Y,u}(1-l_{Y,u}))$, and $z_{u,u'}$ is the inverse of the normalizing constant calculated from the product of two Normals. It is important to note that binomial pdfs are not closed under the product operation. That is, a product of two binomial pdfs are not a binomial pdf unless the binomial prarameters, $l_{X,u}$ and $l_{Y,u}$, are identical. For large n and m, the Normal approximation to Binomial is an important step to maintain the variational structure during the inference procedure.[9] In this way, after eliminating $\mathbf{Y}^m$, the marginal potential $\phi''_{Miid}(\mathbf{X}^n)$ is still represented as the variational form. For potentials with more than two atoms, the same intuition can be applied to shows the property.

Now, we will show that the product of variational forms in Step (iii) can also be represented as a variational form. Suppose that we have two variational potentials $\phi_{Miid}(\mathbf{X}^n)$ $\phi'_{Miid}(\mathbf{X}^n)$. There could be such potentials in the elimination step as shown in Equation (5.8). The product of $\phi_{Miid}(\mathbf{X}^n)$ and $\phi''_{Miid}(\mathbf{X}^n)$ is as follows:

$$\begin{aligned}
&\left(\sum_{u=1}^{k} w_u \cdot f_{\mathcal{B}}(h_x; n, l_{X,u})\right) \cdot \left(\sum_{u'=1}^{k'} w'_{u'} \cdot f'_{\mathcal{B}}(h_x; n, l_{X,u'})\right) \\
&\approx \sum_{u=1}^{k}\sum_{u'=1}^{k'} w_u \cdot w'_{u'} \int f_{\mathcal{N}}(h_x; \mu_u, \sigma_u^2) \cdot f'_{\mathcal{N}}(h_x; \mu_{u'}, \sigma_{u'}^2)\, \mathrm{d}h_x \\
&= \sum_{u=1}^{k}\sum_{u'=1}^{k'} w_u \cdot w'_{u'} \cdot z_{u,u'} f_{\mathcal{N}}(h_x; \mu_{new}, \sigma_{new}^2) = \phi'''_{Miid}(\mathbf{X}^n), \qquad (5.9)
\end{aligned}$$

$z_{u,u'}$ is the inverse of the normalizing constant. This derivation shows that a product of variational potentials results in a variational potential as $\phi'''_{Miid}(\mathbf{X}^n)$.[10]

[9] For small n and m, the binomial distributions can be reversed to the value-histogram representation.

[10] $\phi'''_{Miid}(\mathbf{X}^n)$ is a mixture of $|k \cdot k'|$ Normals. When $|k \cdot k'|$ is large, it is possible to merge some Gaussians.

5.6.2 Inference with Continuous Variables

For continuous variables, we also demonstrate the intuition by an example with two potentials $\phi_{Miid}(\mathbf{X}^n, \mathbf{Y}^m)$ $\phi'_{Miid}(\mathbf{Y}^m)$ where $\mathbf{X}^n$ and $\mathbf{Y}^m$ are two sets of continuous rvs. Each potential will be represented as shown in Section 5.5.2. When we eliminate $\mathbf{Y}^m$ it can be formulated as follows:

$$\begin{aligned}
&\int \left(\sum_{u=1}^{k} w_u \prod_{i=1}^{n} \hat{f}_{l_{X,u}}(X(a_i)) \prod_{j=1}^{m} \hat{f}_{l_{Y,u}}(Y(b_j)) \right) \phi'_{Miid}(\mathbf{Y}^m)\, \mathrm{d}\mathbf{Y} \\
=&\sum_{u=1}^{k} \sum_{u'=1}^{k'} w_u w_{u'} \prod_{i=1}^{n} \hat{f}_{l_{X,u}}(X(a_i)) \prod_{j=1}^{m} \left(\int \hat{f}_{l_{Y,u}}(Y(b_j)) \hat{f}_{l_{Y,u'}}(Y(b_j))\, \mathrm{d}Y(b_j) \right) \\
=&\sum_{u=1}^{k} \sum_{u'=1}^{k'} w_u w_{u'} {z_{u,u'}}^m \prod_{i=1}^{n} \hat{f}_{l_{X,u}}(X(a_i)) = \phi''_{Miid}(\mathbf{X}^n), \qquad (5.10)
\end{aligned}$$

$z_{l,l'}$ is the inverse of the normalizing constant of the product of two mixtures of Normals, $\hat{f}_{l_{Y,u}}(Y(b_j))$ and $\hat{f}_{l_{Y,u'}}(Y(b_j))$.

Finally, we can also show that the product of two variational potentials is a variational potential: $\left(\sum_{u=1}^{k} w_u \cdot \prod_{i=1}^{n} \hat{f}_{l_{X,u}}(X(a_i))\right) \cdot \left(\sum_{u'=1}^{k'} w'_{u'} \cdot \prod_{i=1}^{n} \hat{f'}_{l_{X,u'}}(X(a_i))\right)$

$$\begin{aligned}
&= \sum_{u=1}^{k} \sum_{u'=1}^{k'} w_u w'_{u'} \prod_{i=1}^{n} \hat{f}_{l_{X,u}}(X(a_i)) \hat{f'}_{l_{X,u'}}(X(a_i)) \\
&= \sum_{u=1}^{k} \sum_{u'=1}^{k'} w_u w'_{u'} \cdot {z_{u,u'}}^n \cdot \prod_{u=1}^{n} \hat{f}^{new}_{l_{X,u,u'}}(X(a_i)) = \phi'''_{Miid}(\mathbf{X}^n). \qquad (5.11)
\end{aligned}$$

5.6.3 Lifted Markov chain Monte Carlo (MCMC)

When variational RHMs are still complex for the latent VE, we use a lifted MCMC algorithm, which has following steps: (i) choosing a LV (e.g L_X) to sample; (ii) calculating the conditional probability of the LV (e.g. $\Phi_X(L_X)$) using assignment of neighboring LVs; (iii) choosing an assignment from the distribution (e.g. L_X=l_u ($1 \leq u \leq k$)); and (iv) repeating until convergence.

Here, the main steps are the steps (ii) and (iii). Step (ii) is a subset of the

procedure in Equations (5.8) and (5.10), because the values of neighboring LVs (e.g. $L_Y{=}l_{Y,u'}$ $(1{\leq}u'{\leq}k')$) can be simply assigned instead of summing out (e.g. $\sum_{u'=1}^{k'}$). Step (iii) is a procedure to choose one component based on the weights in the mixture of distributions. For example, $w_u{\cdot}w'_{u'}{\cdot}z_{u,u'}$ in Equation (5.9) is a weight for one of $|k| \cdot |k'|$ Normal distributions in $\phi'''_{Miid}(\mathbf{X}^n)$.

5.7 Relational-Variational Lemmas

For the error analysis, we need to define a term, **$\overline{\mathbf{n}}$-extendible**:

Definition 2 ($\overline{\mathbf{n}}$-extendible) *$P(\mathbf{X}^n)$, a probability with n exchangeable rvs, is* **$\overline{\mathbf{n}}$-extendible** *when the followings hold: (1) there is $P(\mathbf{X}^{\overline{n}})$, a probability with $\overline{n}$ exchangeable rvs ($\overline{n} > n$); and (2) $P(\mathbf{X}^n)$ is the marginal distribution of $P(\mathbf{X}^{\overline{n}})$, i.e., eliminating ($\overline{n} - n$) rvs.*

Figure 5.5 explains the intuition of $\overline{\mathbf{n}}$-extendible potentials for discrete models. If a potential is not extendible, it has no smoothed bars, e.g. a single bar. If a potential is extendible to a large $\overline{n}$, the potential has smoothed bars. If a potential is ∞-extendible, it is exactly a mixture of binomial distributions.

Lemma 12 [Diaconis and Freedman, 1980] *If $P(\mathbf{X}^n)$, a probability with n exchangeable rvs, is $\overline{n}$-extendible, then the Total Variation Distance (***TVD***)*[11] *$||P(\mathbf{X}^n) - P_{Miid}(\mathbf{X}^n)||$ of the input probability $P(\mathbf{X}^n)$ and the variational form $P_{Miid}(\mathbf{X}^n)$ in Equation* (5.2) *is bounded as follows: (i) when $X(a_i)$ are d-valued discrete rvs, the TVD $\leq \frac{2dn}{\overline{n}}$; (ii) when $X(a_i)$ are continuous rvs, the TVD $\leq \frac{n(n-1)}{\overline{n}}$.*

5.7.1 Our Results: Variational RHMs

Factoring Potentials with Multiple Atoms: De Finetti-Heweitt-Savage's theorem of the previous section are applicable only to potentials with a single atom.

[11] The Total Variation Distance (TVD) is $||P - Q|| = sup_{A \in \mathcal{B}}(P(A) - Q(A))$ when $\mathcal{B}$ is a class of Borel sets.

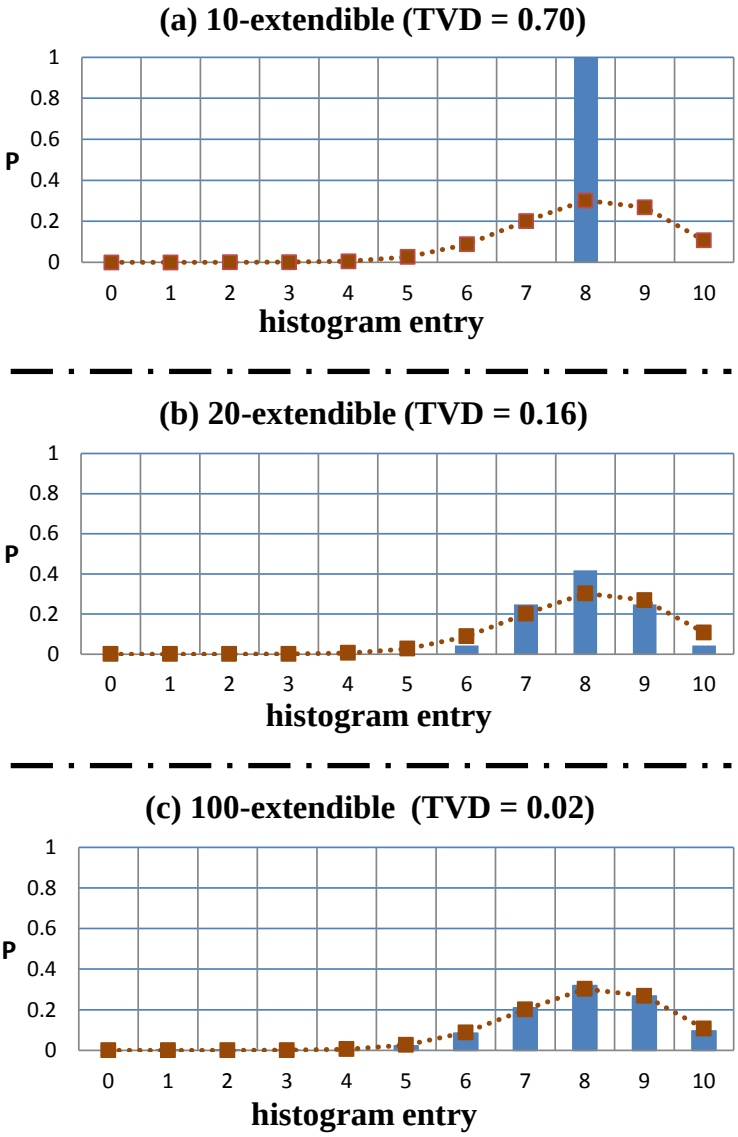


Figure 5.5: Illustrations of three different value-histograms of 10 exchangeable discrete rvs. Dotted lines with markers represent the best possible variational approximation, i.e., the binomial distribution for discrete rvs. (a) presents a potential, which is not extendible to n>10 because of the single bar at 8. (b) and (c) respectively present potentials extendible up to 20 and 100 rvs. For a potential in (1), the variational approximation has a high error, TVD, and thus is not appropriate. When a potential is extendible to a number larger than 10, the variational approximation is reasonably small as shown in (2) and (3).

Here, we present our new theoretical result to construct variational models for RHMs.

Lemma 13 [The Existence of a Variational Form] *For $P(\mathbf{X}^n, \mathbf{Y}^m)$, a probability with two atoms in an RHM, there are two new LVs, L_X and L_Y, and a new potential $\Phi_{XY}(L_X, L_Y)$ such that the following holds,* $\lim_{n,m\to\infty} P(\mathbf{X}^n, \mathbf{Y}^m)$

$$= \int \Phi(L_X, L_Y) \prod_{i=1}^{n} \phi_X(X(a_i)|L_X) \prod_{j=1}^{m} \phi_Y(Y(b_j)|L_Y) \, \mathrm{d}L_X L_Y$$
$$= P_{Miid}(\mathbf{X}^n, \mathbf{Y}^m).$$

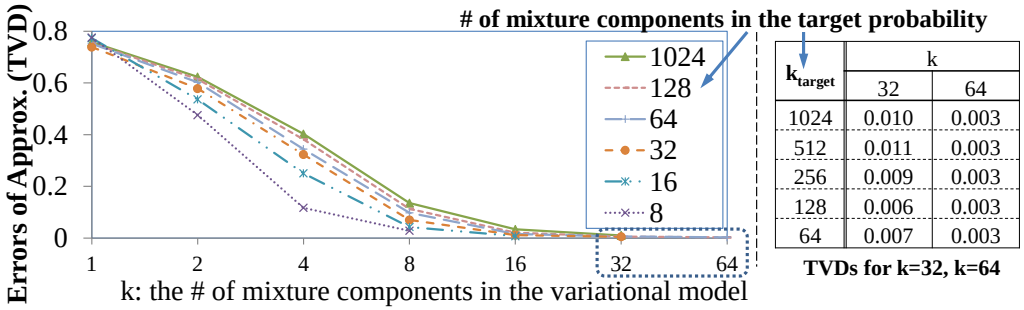
Sketch of proof Assigning values to one atom (e.g. $\mathbf{Y}^m = \mathbf{v}$) results in a new potential with one atom, $\phi(\mathbf{X}^n)$, which can be factored in Equation (5.2). It is not hard to see that $\Phi(L_X|\mathbf{v})$ is conditioned on $\mathbf{v}$. Then, $\Phi(L_X|\mathbf{Y}^m)$ can be factored into $\int \Phi(L_X, L_Y) \prod_j \phi_Y(Y(b_j)|L_Y) \mathrm{d}L_Y$.

To analyze variational error of potential with multiple atoms, we introduce a new term $(\overline{\mathbf{n}}, \overline{\mathbf{m}})$**-extendible**.

Definition 3 ($(\overline{\mathbf{n}}, \overline{\mathbf{m}})$-extendible) *$P(\mathbf{X}^n, \mathbf{Y}^m)$ is $(\overline{\mathbf{n}}, \overline{\mathbf{m}})$-extendible when (1) there is $P(\mathbf{X}^{\overline{n}}, \mathbf{Y}^{\overline{m}})$, a probability with two sets of exchangeable rvs ($\overline{n}$>n, $\overline{m}$>m); and (2) $P(\mathbf{X}^n, \mathbf{Y}^m)$ is the marginal distribution of $P(\mathbf{X}^{\overline{\mathbf{n}}}, \mathbf{Y}^{\overline{\mathbf{m}}})$.*

Lemma 14 (The Error of the Variational Parfactor) *If $P(\mathbf{X}^n, \mathbf{Y}^m)$, a probability with two exchangeable rvs in an RHM, is $(\overline{n}, \overline{m})$-extendible, then the TVD $||P(\mathbf{X}^n, \mathbf{Y}^m) - P_{Miid}(\mathbf{X}^n, \mathbf{Y}^m)||$ is bounded as follows: (i) when $\mathbf{X}^n$ and $\mathbf{Y}^m$ are respectively d_x-valued and d_y-valued discrete rvs, the TVD $\leq \frac{2d_x n}{\overline{n}} + \frac{2d_y m}{\overline{m}}$; (ii) when $\mathbf{X}^n$ are d_x-valued discrete and $\mathbf{Y}^m$ are continuous, the TVD $\leq \frac{2d_x n}{\overline{n}} + \frac{m(m-1)}{\overline{m}}$; (iii) when $\mathbf{X}^n$ and $\mathbf{Y}^m$ are continuous, the TVD $\leq \frac{n(n-1)}{\overline{n}} + \frac{m(m-1)}{\overline{m}}$.*

Sketch of proof The intuition developed on the result in [Diaconis and Freedman, 1980] is that the error of a variational model is additive for an additional atom.



k_{target}	k	
	32	64
1024	0.010	0.003
512	0.011	0.003
256	0.009	0.003
128	0.006	0.003
64	0.007	0.003

TVDs for k=32, k=64

Figure 5.6: The TVD of our variational models with k components. When k is a reasonable size (e.g. 32), the TVD is very small even for a large number of components (e.g. 1024) in the target distributions.

It is straightforward to extend Lemma 14 for probability with more than two sets of rvs, e.g. $P(\mathbf{X}^n, \mathbf{Y}^m, \mathbf{Z}^u)$.

Theorem 15 (*The Error of a Variational RHM*) *Let* $\mathbf{X}_g$ *and* $\mathbf{X}_G$ *is respectively the set of all rvs in a parfactor* g *and an RHM* G. *Let* $P(\mathbf{X}_g)(= \frac{1}{z_g} \prod_{f \in gr(g)} w_f(\mathbf{X}_f))$; *and* $P(\mathbf{X}_G)(= \frac{1}{z} \prod_{g \in G} P(\mathbf{X}_g))$. *The TVD* $\|P(\mathbf{X}_G) - P_{Miid}(\mathbf{X}_G)\|$ *is bounded by the sum of TVD of each parfactor* $\frac{1}{z} \sum_{g \in G} \epsilon_g$ *when* ϵ_g *is* $\left\|P(\mathbf{X}_g) - P_{Miid}(\mathbf{X}_g)\right\|$.

Sketch of proof We can build a fully joint probability of all relational atoms with the RHM. Then, Lemma 14 can be used to prove the TVD of the variational RHM.

5.8 Related Work

[Carbonetto *et al.*, 2005] presents Nonparametric Bayesian Logic (NP-BLOG) which is a variational representation for discrete variables using the Dirichlet Process, a class of nonparametric methods for discrete variables. In principle, the NP-BLOG and the variational RHMs have in common: compact representations for exchangeable rvs. The difference is that NP-BLOG assumes that all discrete, exchangeable rvs are ∞-extendible. Thus, it does not need error analysis or approximations for finite rvs. Here, we investigated further for several new directions: learning algorithms, continuous domains, and approximation errors. Our error bounds for discrete variables provides the better understanding when using

NP-BLOG for a finite number of exchangeable rvs. For example, NP-BLOG has at least the same error when approximating the potential (a) in Figure 5.5.

For discrete cases, exact methods, called value-histogram [de Salvo Braz *et al.*, 2005; Milch *et al.*, 2008], represent potentials using histograms with only relatively small (polynomial) numbers of entries. [Choi *et al.*, 2011a] shows that histogram representations for some special, aggregate, potentials can be approximately replaced by a fixed number of parameters. We generalize and expand the concept to compress general-purpose histogram representations. In addition, we clarify when the variational approximation is good to use and when not.

For continuous potentials, unfortunately, the histogram representation is not applicable because it is not clear how to discretize continuous domains to build up such histograms. Thus, most existing lifted inference for continuous models are limited to Gaussian potentials [Choi *et al.*, 2010a; 2011b; Ahmadi *et al.*, 2011]. Thus, our representation is a unique lifted inference for non-Gaussian continuous potentials.

[Singla and Domingos, 2008] presents the lifted Belief Propagation (BP) by grouping discrete, exchangeable rvs, which send the same messages to neighboring rvs. The intuition, sending the same messages, assumes that the rvs are not constrained among others, thus a single iid not but a mixture of iid rvs. Instead, our lifted MCMC sends a distribution, has more expressive power, and may requires fewer samples until the convergence. We believe that our variational models and the lifted MCMC method can be a good complement to the lifted BP [Singla and Domingos, 2008] for relational models.

When one mixture component of the variational form is given, one may think about the Inversion Elimination [Poole, 2003], especially the Partial Inversion Elimination [de Salvo Braz *et al.*, 2006] because the mixture component is a product of iid rvs. However, existing Inversion Eliminations has specific constraints, and thus not directly applicable to the mixture of iid rvs. In this aspect, our variational models enable applying the Partial Inversion Elimination for lifted inference by the variational learning.

Inferences in hybrid graphical models are proposed, e.g. [Lerner and Parr, 2001; Wang and Domingos, 2008]. [Lerner and Parr, 2001] presents an inference algorithm for Linear-Gaussian Models where linear relations over rvs follow the Gaussian distribution. [Wang and Domingos, 2008] proposed a MCMC algorithm in Hybrid MLNs. However, it converts MLNs into a grounded, or propositionalized, one while sampling.

5.9 Experimental Results

We provide experimental results regarding the variational errors and the efficiency and the accuracy of the LRVI in a real-world groundwater model.

First, we examine the error caused by k, the finite numbers of mixture components. We assume that the target probabilities are ∞-extendible, so that the TVD, variational error, can converge to 0 when k is large enough. $P(\mathbf{X}^{100})$ each target probability over 100 exchangeable binary rvs is a mixture of binomials with various k from 8 to 1024. For each target probability, our EM algorithm (Section 5.5) incrementally learn the variational model, k and ϕ_{Miid} with the EM algorithm. Then, we measure the average TVD, variational error. Figure 5.6 shows the TVD of our EM algorithm. It shows that the TVD becomes reasonably small (≤ 0.01) with only 32 components even for target densities with 1024 components. When we increase the number of rvs to 1000 $P(\mathbf{X}^{1000})$, the results are consistent. Thus, the results show that the finite number of k is not an issues when the variational approximation is applicable, i.e. $\overline{n}$-extendible and $\overline{n} \gg n$.

Second, we apply our variational learning algorithm to a real-world groundwater dataset, Figure 5.7 Republican River Compact Administration (RRCA) , which is composed of measurements at over 10,000 wells and baseflow observations at 65 gages from 1918 until 2007.[12] After calibration, the training dataset is a set of partial continuous observations in a 480 (months) by 3420 (wells) matrix. First,

[12]Head predictions are available via the RRCA official website, http://www.republicanrivercompact.org.

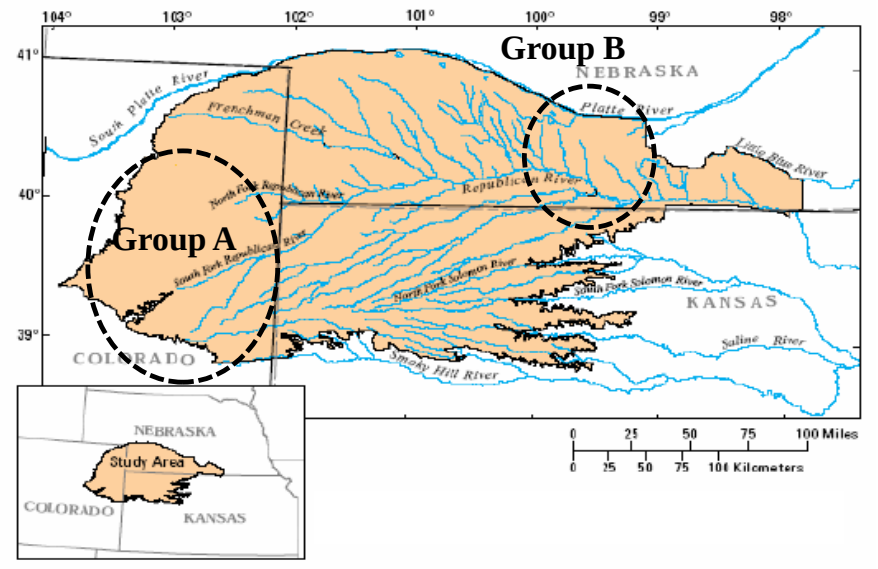


Figure 5.7: Locations of clustered wells A and B in the RRCA dataset.

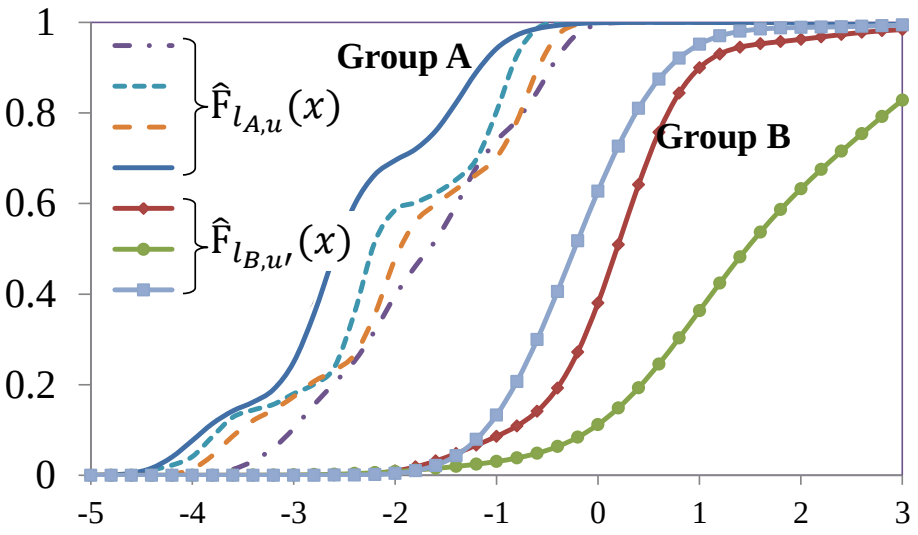


Figure 5.8: Learned empirical distributions, Cdfs ($\hat{F}_{l_{A,u}}(x)$ and $\hat{F}_{l_{B,u'}}(x)$), of rvs in groups A and B.

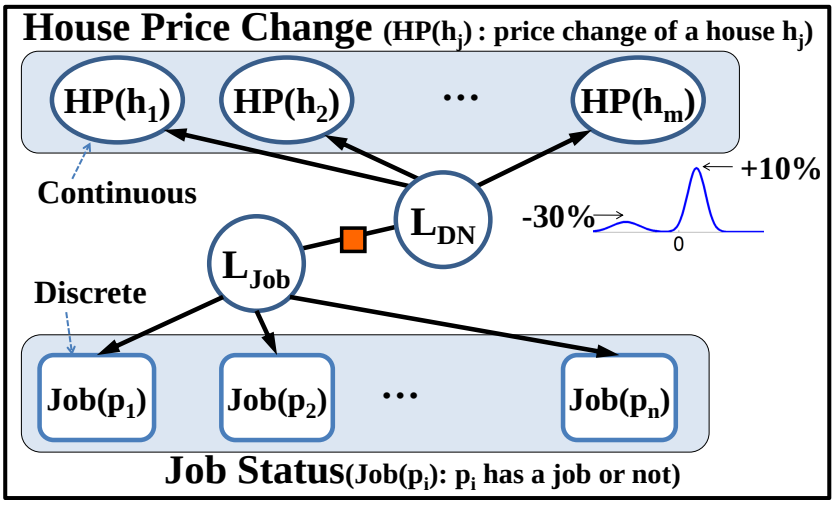
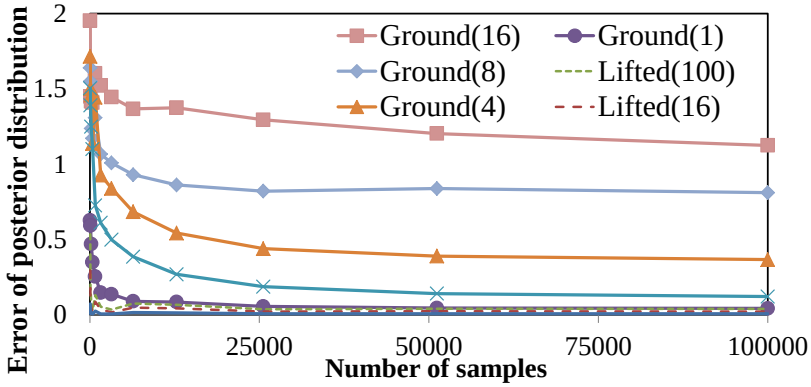


Figure 5.9: A factored variational model for a continuous atom $\mathbf{HP}^m$, the price change of each house, and a discrete atom $\mathbf{Job}^n$, whether each individual has a job.

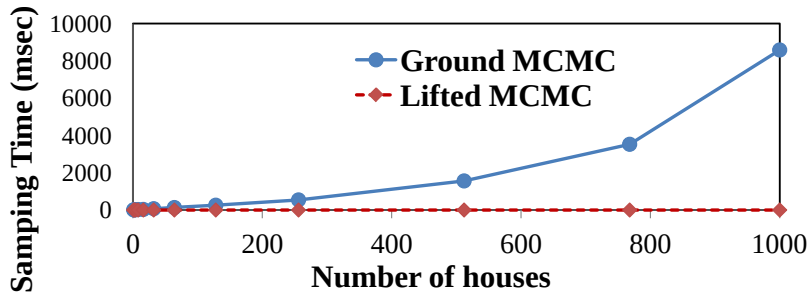
we cluster the 3420 wells into 10 groups which show similar observations, means and variances, (approximately exchangeable).[13] From the dataset, the EM algorithm directly learns a variational model until the log-likelihood converges. As a result, from 6 to 14 mixtures of Gaussians (MoGs) are learned for each cluster. Figure 5.7 shows some learned empirical distributions, cdfs of MoGs, with high weights from two clustered area, A and B. To represent the joint distribution over the clusters, we convert the 480 by 3240 input matrix into a 480 (months) by 92 (MoGs) matrix.

For each test month, we compute the empirical distribution of the query variables given the partial observations. Our lifted inference returns queries very efficiently (average 0.3 secs) compared to the ground inference (average 37.9 secs). As one expects, ground and variational inference tasks use different sizes of matrices 480 by 3420 and 480 by 92 respectively. That explains the reason why the variational inference is efficient. The average TVDs are 0.35 (ground) and 0.29 (variational, better). We conjecture that the relatively high TVDs are from noises of the high-dimensional dataset.

[13] We know that the approximate clustering does not guarantee finding a pure relational model. However, our focus here is to measure the computational efficiency of our variational method.



(a) The accuracy of sampling



(b) The average time to sample

Figure 5.10: Figure (a) compares the accuracy of our lifted MCMC and the ground MCMC with various numbers of houses. '()' indicates the number of houses (e.g. 'Ground(16)' is the ground MCMC with 16 houses, and 'Lifted(16)' is the lifted MCMC with 16 houses). Figure (b) shows the average sampling time per each time step with various number of houses.

Third, we compare the accuracy and the efficiency of our lifted MCMC algorithm with a vanilla (ground) MCMC algorithm on an already factored variational model. The model is composed of two relational atoms: a binary atom $\mathbf{Job}^n$, saying whether each individual has a job, and a continuous atom $\mathbf{HP}^m$, saying the price change of each house. L_{Job} is a latent variable, which represents the Bernoulli parameters of $\phi_{Miid}(\mathbf{Job}^n)$. L_{DN} is a latent variable, which represents the mixture of two Gaussians,[14]

$$\phi_{Miid}(\mathbf{HP}^m) = L_{DN} \prod_{j=1}^{m} f_{\mathcal{N}}(HP(h_j); -0.3, \sigma^2_{DN}) + (1 - L_{DN}) \prod_{j=1}^{m} f_{\mathcal{N}}(HP(h_j); 0.1, \sigma^2_{UP}).$$

Then, the potential between two latent variables follows a linear Gaussian:

$$\Phi(L_{Job}, L_{DN}) = N(L_{Job} - L_{DN}, \sigma^2_{JH}).$$

Figure 5.10 (a) shows the accuracy of the two algorithms given the same number of samples. That is, it measure the ratio of error to estimate a probability of a randomly chosen variable x, $|p_{true}(x) - p_{MCMC}(x)|/p_{true}(x)$. It shows that our lifted MCMC converges to the true density much faster than the ground MCMC. Figure 5.10 (b) shows the average sampling time (per step) with different number of RVS (e.g. the number of houses).

5.10 Conclusion and Future Work

We propose new lifted relational variational inference algorithms for relational hybrid models. Our main contributions are two folds: (1) in theory, we show that a relational model, which can represent large-scale systems, is accurately represented by a variational model; (2) our lifted algorithms are the first to solve infer-

[14] Because the target distribution is a mixture of two iid Gaussians, the lifted Belief Propagation (BP) [Singla and Domingos, 2008] is not applicable without a modification. Although we can modify it for continuous domain, the lifted BP assumes that each house sends the same message,i.e. price. Here, the messages between houses should be constrained among others. That is, the prices of some houses go down, and then prices of the other houses should go up.

ence problems without referring ground rvs for non-Gaussian continuous models. Experiments show that our algorithm outperforms the existing possible methods in a real-world problem.

There are some limitations of this work. First, for continuous potentials, the variational learning may require extensive computations. Especially, when there is no analytic solution, we may need (up to) $O(exp(n))$ samples. Unfortunately, this is a hard problem in general. Finding variational forms for well-known distributions, such as logistic regression [Doucet *et al.*, 2000], would be interesting future research. Second, clustering continuous exchangeable rvs is still unsolved problem because of the noise in continuous data. We only know solutions for discrete domains [Kok and Domingos, 2008; 2009]. However, the methods are not directly applicable to continuous domains due to the noise.

CHAPTER 6

SUMMARY AND FUTURE WORK

This thesis presents efficient methods to answer questions on large-scale probabilistic graphical models, specifically Relational Hybrid Models (RHMs). The key contributions of this thesis are two folds: (1) novel lifted inference algorithms for large-scale probabilistic graphical models with continuous variables (Chapter 2 and 3); and (2) new insights that transform probabilistic first-order languages into compact approximations which allow efficient ways of inference (Chapter 4 and 5).

6.1 Summary of Contributions

This thesis offers the following contributions:

- It introduces new lifted inference algorithms that compute conditional and marginal probabilities of probabilistic graphical models with continuous variables, Relational Continuous Models (RCMs), especially the pairwise linear Gaussian (Chapter 2) [Choi *et al.*, 2010a];
- It presents a new exact Kalman filter (KF), the Lifted Relational Kalman filter (LRKF), that enables scaling the exact Kalman filter from thousands of variables to billions of variables (Chapter 3) [Choi *et al.*, 2011b];
- It shows that typically used aggregate operators over Probabilistic Relational Graphical Models (PRGMs) and the existential quantification over probabilistic first-order logic can be accurately approximated by linear constraints in Gaussian distributions. (Chapter 4) [Choi *et al.*, 2011a];

- It proves that Relational Hybrid Models, relational models with continuous and discrete domains, can be accurately approximated by variational models, composed of mixtures of independent and identically distributed(i.i.d.) random variables. The approximation allows efficient inference procedures on the large-scale probabilistic graphical models with hybrid domains (Chapter 5) [Choi and Amir, 2011; 2012].

Chapter 2 introduces new lifted inference algorithms that compute conditional and marginal probabilities of RCMs, large probabilistic graphical models with continuous variables only. The algorithm is a key advance in exact inference for RCMs, since most previous works are restricted to discrete domains. Given a query and a set of observations, the algorithm exactly computes the conditional probability of the query, when potentials satisfy specified conditions in the chapter. The algorithm maintains relational structures during the inference procedure for relational pair-wise potentials such as pairwise linear Gaussian potentials.

The Kalman filter (KF) is a computational tool with widespread applications in robotics, financial and weather forecasting, environmental engineering and defense. Given observation and state transition models, the KF recursively estimates the state variables of a dynamic system. Chapter 3 presents a new efficient filtering algorithm, the LRKF, for large-scale linear dynamic systems. The LRKF extends the RCMs with pairwise linear Gaussian potentials in Chapter 2, and represents dynamic systems in a relational (first-order) way. In each time step, the lifted inference algorithm efficiently updates the large number of random variables at the first-order (relational) level. The LRKF maintains compact pairwise relationships among random variables even with individual observations. Thus, individual attributes do not degenerate the relational structures into propositional ones. Theoretical analysis and empirical tests show that this approach leads to significant gains in efficiency and enables filtering for systems with very large numbers of random variables.

Processing aggregate parfactors efficiently is a fundamental problem since they involve functions commonly used in writing models. Chapter 4 adds efficient exact methods for the binary case k=2, as well as efficient approximations for the cases in which sets of aggregated variables are large, which is precisely the situation in which we are more likely to use aggregate factors in the first place. The chapter shows that aggregate operators over relational models can be accurately approximated by linear constraints over Gaussian distributions. Thus, in many case, calculating the conditional probability does not depend on the number of random variables. The accuracy of approximation is close to optimal when the model has a large number of random variables. The approximation will therefore be an important part of practical applications of relational graphical models.

Finally, Chapter 5 presents new understandings that relate RHMs and variational models. The contributions of this chapter are two folds: (1) in theory, it shows that a relational model, that can represent large-scale systems, is accurately represented by a variational model; (2) the lifted variational inference algorithms are the first to solve inference problems without referring to ground random variables for non-Gaussian continuous models. The variational models also represent the discrete and continuous variables in a coherent way. RHMs are general, so that the new insights can be applied to other models such as Markov Logic Networks (MLNs) and Latent Tree Models (LTMs). Experiments show that the algorithm outperforms the existing methods in a real-world problem.

6.2 Future Work

Most Relational Models assume that each set of random variables has the same types of relationships with other sets. Thus, whenever an individual attribute is assigned or observed to a random variable, the random variable is separated from the relational structure. This thesis shows that it is possible to maintain relational structures even if individual attributes are made to large linear dynamic systems

with continuous variable (Chapter 3) and to large-scale graphical models with both discrete and continuous domains by approximations (Chapter 5). However, in discrete models or non-Gaussian models, individual attributes prevent exact lifted inference algorithms from maintaining relational structures. Thus, handling individual attributes for Relational Models with discrete domains would be a promising research direction.

Another key challenge is to learn RPGMs from noisy data. Building a relational model by clustering continuous random variables is hard especially for noisy continuous inputs. Suppose that a house X is on the border of two cities A and B. A random variable representing the price of the house X would have similar relationships with random variables for housing prices both in A and random variable in B. It is not clear that a random variable should be included either one cluster or another. In this reason, solutions for discrete domains [Kok and Domingos, 2008; 2009] are not directly applicable to continuous models. One possible solution is building several Relational Models with different granularities [de Salvo Braz *et al.*, 2009; Kiddon and Domingos, 2011]. Finding more principles to cluster continuous random variables may pave the way for new applications, such in econometrics, computer vision, robot planning, and environmental engineering.

One limitation of the LRKF is that it shatters the model when the random variables in a relational atom receive different numbers of observations because their variances and covariances become different. Approximate re-grouping of random variables would be a general recourse in this case. Although the Gaussian distribution has widespread use in real-world applications, the linearity assumption in the KF limits the use of the Relational Kalman filter to non-linear dynamic systems. Thus, understanding principles for non-linear dynamic systems would be crucial to building a new relational Extended Kalman filter (EKF). In the general case, building a relational stochastic filter, e.g. Particle filter [Doucet *et al.*, 2000], could be an attractive alternative.

REFERENCES

[Ahmadi *et al.*, 2011] Babak Ahmadi, Kristian Kersting, and Scott Sanner. Multi-evidence lifted message passing, with application to pagerank and the kalman filter. In Walsh [2011], pages 1152–1158.

[Apsel and Brafman, 2011] Udi Apsel and Ronen I. Brafman. Extended lifted inference with joint formulas. In *Proceedings of the Twenty-Seventh Conference on Uncertainty in Artificial Intelligence, UAI 2011*, pages 11–18, 2011.

[Bacchus, 1990] Fahiem Bacchus. *Representing and reasoning with probabilistic knowledge: a logical approach to probabilities*. MIT Press, Cambridge, MA, USA, 1990.

[Bahmani-Oskooee and Brown, 2004] Mohsen Bahmani-Oskooee and Ford Brown. Kalman filter approach to estimate the demand for international reserves. *Applied Economics*, 36(15):1655–1668, 2004.

[Boutilier, 2009] Craig Boutilier, editor. *IJCAI 2009, Proceedings of the 21st International Joint Conference on Artificial Intelligence, Pasadena, California, USA, July 11-17, 2009*. IJCAI, 2009.

[Burgard and Roth, 2011] Wolfram Burgard and Dan Roth, editors. *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2011, San Francisco, California, USA, August 7-11, 2011*. AAAI Press, 2011.

[Burgers *et al.*, 1998] Gerrit Burgers, Peter Jan van Leeuwen, Geir Evensen, Gerrit Burgers, and Gerrit Burgers. On the analysis scheme in the ensemble kalman filter. *Monthly Weather Review*, 126:1719–1724, 1998.

[Carbonetto *et al.*, 2005] Peter Carbonetto, Jacek Kisynski, Nando de Freitas, and David Poole. Nonparametric bayesian logic. In Fahiem Bacchus and Tommi Jaakkola, editors, *Proceedings of the 21st Conference in Uncertainty in Artificial Intelligence, UAI 2005*, pages 85–93. AUAI Press, 2005.

[Choi and Amir, 2007] Jaesik Choi and Eyal Amir. Factor-guided motion planning for a robot arm. In *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2007*, pages 27–32. IEEE, 2007.

[Choi and Amir, 2009] Jaesik Choi and Eyal Amir. Combining planning and motion planning. In *2009 IEEE International Conference on Robotics and Automation, ICRA 2009*, pages 238–244. IEEE, 2009.

[Choi and Amir, 2011] Jaesik Choi and Eyal Amir. Lifted variational inference. Computer science research and tech reports, University of Illinois, 2011.

[Choi and Amir, 2012] Jaesik Choi and Eyal Amir. Lifted relational variational inference. In *Submitted for publication*, 2012.

[Choi *et al.*, 2008] Jaesik Choi, Won J. Jeon, and Sang-Chul Lee. Spatio-temporal pyramid matching for sports videos. In Michael S. Lew, Alberto Del Bimbo, and Erwin M. Bakker, editors, *Multimedia Information Retrieval, MIR 2008*, pages 291–297. ACM, 2008.

[Choi *et al.*, 2010a] Jaesik Choi, Eyal Amir, and David Hill. Lifted inference for relational continuous models. In Peter Grünwald and Peter Spirtes, editors, *Proceedings of the Twenty-Sixth Conference on Uncertainty in Artificial Intelligence, UAI 2010*, pages 126–134. AUAI Press, 2010.

[Choi *et al.*, 2010b] Jaesik Choi, Jay Pujara, Vishwanth Tumkur Ramarao, and Ke Wei. Identifying ip addresses for spammers. *U.S. Patent No. 7,849,146*, 2010.

[Choi *et al.*, 2011a] Jaesik Choi, Rodrigo de Salvo Braz, and Hung H. Bui. Efficient methods for lifted inference with aggregate factors. In Burgard and Roth [2011].

[Choi *et al.*, 2011b] Jaesik Choi, Abner Guzman-Rivera, and Eyal Amir. Lifted relational kalman filtering. In Walsh [2011], pages 2092–2099.

[Choi *et al.*, 2011c] Jaesik Choi, Ke Wei, and Vishwanath Tumkur Ramarao. Filter for blocking image-based spam. *U.S. Patent No. 8,055,078*, 2011.

[Choi *et al.*, 2011d] Myung Jin Choi, Vincent Y. F. Tan, Animashree Anandkumar, and Alan S. Willsky. Learning latent tree graphical models. *Journal of Machine Learning Research*, 12:1771–1812, 2011.

[Chu *et al.*, 2006] Wei Chu, Vikas Sindhwani, Zoubin Ghahramani, and S. Sathiya Keerthi. Relational learning with gaussian processes. In Bernhard Schölkopf, John C. Platt, and Thomas Hoffman, editors, *Proceedings of the Twentieth Annual Conference on Neural Information Processing Systems, NIPS 2006*, pages 289–296. MIT Press, 2006.

[Cowell, 1998] Robert Cowell. Advanced inference in bayesian networks. In Michel I. Jordan, editor, *Learning in graphical models*, pages 27–49. MIT Press, Cambridge, MA, 1998.

[Damien and Walker, 2001] Paul Damien and Stephen G Walker. Sampling truncated normal, beta, and gamma densities. *Journal of Computational and Graphical Statistics*, 10(2):206–215, 2001.

[Dasgupta, 1999] Sanjoy Dasgupta. Learning mixtures of gaussians. In *The 40th Annual Symposium on Foundations of Computer Science, FOCS 1999*, pages 634–644. IEEE Computer Society, 1999.

[de Finetti, 1931] B. de Finetti. Funzione caratteristica di un fenomeno aleatorio. *Mathematice e Naturale*, 1931.

[de Salvo Braz *et al.*, 2005] Rodrigo de Salvo Braz, Eyal Amir, and Dan Roth. Lifted first-order probabilistic inference. In Kaelbling and Saffiotti [2005], pages 1319–1325.

[de Salvo Braz *et al.*, 2006] Rodrigo de Salvo Braz, Eyal Amir, and Dan Roth. Mpe and partial inversion in lifted probabilistic variable elimination. In *Proceedings of the Twenty-First National Conference on Artificial Intelligence and the Eighteenth Innovative Applications of Artificial Intelligence Conference, AAAI/IAAI 2006*, pages 1123–1130. AAAI Press, 2006.

[de Salvo Braz *et al.*, 2007] R. de Salvo Braz, E. Amir, and D. Roth. Lifted first-order probabilistic inference. In L. Getoor and B. Taskar, editors, *An Introduction to Statistical Relational Learning*, pages 433–451. MIT Press, 2007.

[de Salvo Braz *et al.*, 2009] R. de Salvo Braz, S. Natarajan, H. Bui, J. Shavlik, and S. Russell. Anytime lifted belief propagation. In *Statistical Relational Learning Workshop*, 2009.

[Diaconis and Freedman, 1980] P. Diaconis and D. Freedman. Finite exchangeable sequences. *Annals of Probability*, 8(1):115–130, 1980.

[Diaconis, 1977] P. Diaconis. Finite forms of de finetti's theorem on exchangeability. *Synthese*, 36(2):271–281, 1977.

[Díez and Galán, 2003] F. J. Díez and S. F. Galán. Efficient computation for the noisy MAX. *International Journal of Approximate Reasoning*, 18:165–177, 2003.

[Doucet *et al.*, 2000] Arnaud Doucet, Nando de Freitas, Kevin P. Murphy, and Stuart J. Russell. Rao-blackwellised particle filtering for dynamic bayesian networks. In Craig Boutilier and Moisés Goldszmidt, editors, *Proceedings of the 16th Conference in Uncertainty in Artificial Intelligence, UAI 2000*, pages 176–183. Morgan Kaufmann, 2000.

[Esseen, 1942] Carl-Gustav Esseen. On the liapunoff limit of error in the theory of probability. *Arkiv foer Matematik, Astronomi, och Fysik*, A28(9):1–19, 1942.

[Evensen, 1994] Geir Evensen. Sequential data assimilation with a nonlinear quasi-geostrophic model using monte carlo methods to forecast error statistics. *Journal of Geophysical Research*, 99:10143–10162, 1994.

[Fox and Gomes, 2008] Dieter Fox and Carla P. Gomes, editors. *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence, AAAI 2008, Chicago, Illinois, USA, July 13-17, 2008*. AAAI Press, 2008.

[Friedman *et al.*, 1999] Nir Friedman, Lise Getoor, Daphne Koller, and Avi Pfeffer. Learning probabilistic relational models. In Thomas Dean, editor, *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence, IJCAI 1999*, pages 1300–1309. Morgan Kaufmann, 1999.

[Getoor and Taskar, 2007] Lise Getoor and Ben Taskar. *Introduction to Statistical Relational Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2007.

[Geweke, 1991] John Geweke. Efficient simulation from the multivariate normal and student-t distributions subject to linear constraints and the evaluation of constraint probabilities. In *Proceedings the 23rd Symposium on the Interface between Computer Sciences and Statistics*, pages 571–578, 1991.

[Gottlob and Walsh, 2003] Georg Gottlob and Toby Walsh, editors. *IJCAI 2003, Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence, Acapulco, Mexico, August 9-15, 2003*. Morgan Kaufmann, 2003.

[Gotze, 1991] F. Gotze. On the rate of convergence in the multivariate clt. *The Annals of Probability*, 19(2):724–739, 1991.

[Hajishirzi *et al.*, 2009] Hannaneh Hajishirzi, Afsaneh Shirazi, Jaesik Choi, and Eyal Amir. Greedy algorithms for sequential sensing decisions. In Boutilier [2009], pages 1908–1915.

[Halpern, 1990] Joseph Y. Halpern. An analysis of first-order logics of probability. *Artif. Intell.*, 46(3):311–350, 1990.

[Hewitt and Savage, 1955] E. Hewitt and L.J. Savage. Symmetric measures on cartesian products. *Trans. Amer. Math. Soc.*, 80:470–501, 1955.

[Hill *et al.*, 2009] David J. Hill, Barbara S. Minsker, and Eyal Amir. Real-time bayesian anomaly detection in streaming environmental data. *Water Resources Research*, 45:W00D28, 2009.

[Jha *et al.*, 2010] Abhay Kumar Jha, Vibhav Gogate, Alexandra Meliou, and Dan Suciu. Lifted inference seen from the other side : The tractable features. In John D. Lafferty, Christopher K. I. Williams, John Shawe-Taylor, Richard S. Zemel, and Aron Culotta, editors, *Proceedings of the 24th Annual Conference on Neural Information Processing Systems, NIPS 2010*, pages 973–981. Curran Associates, Inc., 2010.

[John and Langley, 1995] George H. John and Pat Langley. Estimating continuous distributions in bayesian classifiers. In Philippe Besnard and Steve Hanks, editors, *Proceedings of the Eleventh Annual Conference on Uncertainty in Artificial Intelligence, UAI 1995*, pages 338–345. Morgan Kaufmann, 1995.

[Kaelbling and Saffiotti, 2005] Leslie Pack Kaelbling and Alessandro Saffiotti, editors. *IJCAI 2005, Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence, Edinburgh, Scotland, UK, July 30-August 5, 2005*. Professional Book Center, 2005.

[Kalman, 1960] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, 82(Series D):35–45, 1960.

[Kersting *et al.*, 2006] K. Kersting, L. De Raedt, and T. Raiko. Logical hidden markov models. *Journal of Artificial Intelligence Research*, 25:425–456, 2006.

[Kiddon and Domingos, 2011] Chloe Kiddon and Pedro Domingos. Coarse-to-fine inference and learning for first-order probabilistic models. In Burgard and Roth [2011].

[Kisynski and Poole, 2009] Jacek Kisynski and David Poole. Lifted aggregation in directed first-order probabilistic models. In Boutilier [2009], pages 1922–1929.

[Kok and Domingos, 2008] Stanley Kok and Pedro Domingos. Extracting semantic networks from text via relational clustering. In Walter Daelemans, Bart Goethals, and Katharina Morik, editors, *Machine Learning and Knowledge Discovery in Databases, European Conference, ECML/PKDD 2008*, volume 5211 of *Lecture Notes in Computer Science*, pages 624–639. Springer, 2008.

[Kok and Domingos, 2009] Stanley Kok and Pedro Domingos. Learning markov logic network structure via hypergraph lifting. In Andrea Pohoreckyj Danyluk, Léon Bottou, and Michael L. Littman, editors, *Proceedings of the 26th Annual International Conference on Machine Learning, ICML 2009*, volume 382 of *ACM International Conference Proceeding Series*, page 64. ACM, 2009.

[Koller and Pfeffer, 1997] Daphne Koller and Avi Pfeffer. Object-oriented bayesian networks. In Dan Geiger and Prakash P. Shenoy, editors, *Proceedings of the Thirteenth Conference on Uncertainty in Artificial Intelligence, UAI 1997*, pages 302–313. Morgan Kaufmann, 1997.

[Kotz *et al.*, 2000] S. Kotz, N Balakrishnan, and N.L. Johnson. *Continuous Multivariate Distributions*. Wiley, New York, 2000.

[Lerner and Parr, 2001] Uri Lerner and Ronald Parr. Inference in hybrid networks: Theoretical limits and practical algorithms. In Jack S. Breese and Daphne Koller, editors, *Proceedings of the 17th Conference in Uncertainty in Artificial Intelligence, UAI 2001*, pages 310–318. Morgan Kaufmann, 2001.

[Limketkai *et al.*, 2005] Benson Limketkai, Lin Liao, and Dieter Fox. Relational object maps for mobile robots. In Kaelbling and Saffiotti [2005], pages 1471–1476.

[Mihalkova and Mooney, 2009] Lilyana Mihalkova and Raymond J. Mooney. Transfer learning from minimal target data by mapping across relational domains. In Boutilier [2009], pages 1163–1168.

[Milch and Russell, 2006] Brian Milch and Stuart J. Russell. First-order probabilistic languages: Into the unknown. In Stephen Muggleton, Ramón P. Otero, and Alireza Tamaddoni-Nezhad, editors, *Inductive Logic Programming, 16th International Conference, ILP 2006*, volume 4455 of *Lecture Notes in Computer Science*, pages 10–24. Springer, 2006.

[Milch *et al.*, 2005] Brian Milch, Bhaskara Marthi, Stuart J. Russell, David Sontag, Daniel L. Ong, and Andrey Kolobov. Blog: Probabilistic models with unknown objects. In Kaelbling and Saffiotti [2005], pages 1352–1359.

[Milch *et al.*, 2008] Brian Milch, Luke S. Zettlemoyer, Kristian Kersting, Michael Haimes, and Leslie Pack Kaelbling. Lifted probabilistic inference with counting formulas. In Fox and Gomes [2008], pages 1062–1068.

[Ng and Subrahmanian, 1992] Raymond Ng and V. S. Subrahmanian. Probabilistic logic programming. *Information and Computation*, 101(2):150–201, 1992.

[Niemira and Saaty, 2004] Michael P. Niemira and Thomas L. Saaty. An analytic network process model for financial-crisis forecasting. *International Journal of Forecasting*, 20(4):573–587, 2004.

[Paskin, 2003] Mark A. Paskin. Thin junction tree filters for simultaneous localization and mapping. In Gottlob and Walsh [2003], pages 1157–1166.

[Pfeffer *et al.*, 1999] Avi Pfeffer, Daphne Koller, Brian Milch, and Ken T. Takusagawa. Spook: A system for probabilistic object-oriented knowledge representation. In Kathryn B. Laskey and Henri Prade, editors, *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence, UAI 1999*, pages 541–550. Morgan Kaufmann, 1999.

[Poole, 2003] David Poole. First-order probabilistic inference. In Gottlob and Walsh [2003], pages 985–991.

[Rasmussen and Ghahramani, 2001] Carl Edward Rasmussen and Zoubin Ghahramani. Infinite mixtures of gaussian process experts. In Thomas G. Dietterich, Suzanna Becker, and Zoubin Ghahramani, editors, *Advances in Neural Information Processing Systems 14, Neural Information Processing Systems: Natural and Synthetic*, pages 881–888. MIT Press, 2001.

[Rice, 2006] John A. Rice. *Mathematical Statistics and Data Analysis*. Duxbury Press, 2006.

[Richardson and Domingos, 2006] Matthew Richardson and Pedro Domingos. Markov logic networks. *Machine Learning*, 62(1-2):107–136, 2006.

[Roweis and Ghahramani, 1999] Sam Roweis and Zoubin Ghahramani. A unifying review of linear gaussian models. *Neural Computation*, 11(2):305–345, 1999.

[Rue and Held, 2005] H. Rue and L. Held. *Gaussian Markov random fields: Theory and applications*. Springer, New York, 2005.

[Shenoy, 2006] Prakash P. Shenoy. Inference in hybrid bayesian networks using mixtures of gaussians. In Rina Dechter and Thomas Richardson, editors, *Proceedings of the 22nd Conference in Uncertainty in Artificial Intelligence, UAI 2006*. AUAI Press, 2006.

[Singla and Domingos, 2007] Parag Singla and Pedro Domingos. Markov logic in infinite domains. In Ronald Parr and Linda C. van der Gaag, editors, *Proceedings of the Twenty-Third Conference on Uncertainty in Artificial Intelligence, UAI 2007*, pages 368–375. AUAI Press, 2007.

[Singla and Domingos, 2008] Parag Singla and Pedro Domingos. Lifted first-order belief propagation. In Fox and Gomes [2008], pages 1094–1099.

[Sorenson and Stubberud, 1968] H. W. Sorenson and A. R. Stubberud. Nonlinear filtering by approximation of the a posteriori density. *International Journal of Control*, 8:33–51, 1968.

[Walsh, 2011] Toby Walsh, editor. *IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence, Barcelona, Catalonia, Spain, July 16-22, 2011*. IJCAI/AAAI, 2011.

[Wang and Domingos, 2008] Jue Wang and Pedro Domingos. Hybrid markov logic networks. In Fox and Gomes [2008], pages 1106–1111.

[Xu and Jordan, 1995] Lei Xu and Michael I. Jordan. On convergence properties of the em algorithm for gaussian mixtures. *Neural Computation*, 8:129–151, 1995.

[Xu *et al.*, 2009] Zhao Xu, Kristian Kersting, and Volker Tresp. Multi-relational learning with gaussian processes. In Boutilier [2009], pages 1309–1314.

[Xu *et al.*, 2012] Tiangfang Xu, Albert J. Valocchi, Jaesik Choi, and Eyal Amir. Improving groundwater flow model prediction using complementary data-driven models. In *XIX International Conference on Computational Methods in Water Resources, CMWR 2012*, 2012.

[Zhang, 2002] Nevin Lianwen Zhang. Hierarchical latent class models for cluster analysis. In Rina Dechter and Richard S. Sutton, editors, *Proceedings of the Eighteenth National Conference on Artificial Intelligence and Fourteenth Conference on Innovative Applications of Artificial Intelligence, AAAI/IAAI 2002*, pages 230–237. AAAI Press / The MIT Press, 2002.